# Bayesian Time Series Models and Scalable Inference

by

Matthew James Johnson
B.S. Electrical Engineering and Computer Sciences, UC Berkeley, 2008
S.M. Electrical Engineering and Computer Science, MIT, 2010

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

Doctor of Philosophy
in
Electrical Engineering and Computer Science
at the Massachusetts Institute of Technology

June 2014





Signature redacted

Signature of Author: _____

Department of Electrical Engineering and Computer Science
May 21 2014

Signature redacted

Certified by: _____

Alan S. Willsky
Edwin Sibley Webster Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Signature redacted

Accepted by: _____

Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

# Bayesian Time Series Models and Scalable Inference

by Matthew James Johnson
B.S. Electrical Engineering and Computer Sciences, UC Berkeley, 2008
S.M. Electrical Engineering and Computer Science, MIT, 2010

Submitted to the Department of Electrical Engineering
and Computer Science on May 21 2014
in Partial Fulfillment of the Requirements for the Degree
of Doctor of Philosophy in Electrical Engineering and Computer Science

## Abstract

With large and growing datasets and complex models, there is an increasing need for scalable Bayesian inference. We describe two lines of work to address this need.

In the first part, we develop new algorithms for inference in hierarchical Bayesian time series models based on the hidden Markov model (HMM), hidden semi-Markov model (HSMM), and their Bayesian nonparametric extensions. The HMM is ubiquitous in Bayesian time series models, and it and its Bayesian nonparametric extension, the hierarchical Dirichlet process hidden Markov model (HDP-HMM), have been applied in many settings. HSMMs and HDP-HSMMs extend these dynamical models to provide state-specific duration modeling, but at the cost of increased computational complexity for inference, limiting their general applicability. A challenge with all such models is scaling inference to large datasets.

We address these challenges in several ways. First, we develop classes of duration models for which HSMM message passing complexity scales only linearly in the observation sequence length. Second, we apply the stochastic variational inference (SVI) framework to develop scalable inference for the HMM, HSMM, and their nonparametric extensions. Third, we build on these ideas to define a new Bayesian nonparametric model that can capture dynamics at multiple timescales while still allowing efficient and scalable inference.

In the second part of this thesis, we develop a theoretical framework to analyze a special case of a highly parallelizable sampling strategy we refer to as Hogwild Gibbs sampling. Thorough empirical work has shown that Hogwild Gibbs sampling works very well for inference in large latent Dirichlet allocation models (LDA), but there is little theory to understand when it may be effective in general. By studying Hogwild Gibbs applied to sampling from Gaussian distributions we develop analytical results as well as a deeper understanding of its behavior, including its convergence and correctness in some regimes.

---

Thesis Supervisor: Alan S. Willsky
Professor of Electrical Engineering and Computer Science

# Contents

Chapter 1

# Introduction

As datasets grow both in size and in complexity, there is an increasing need for rich, flexible models that provide interpretable structure, useful prior controls, and explicit modeling of uncertainty. The Bayesian paradigm provides a powerful framework for approaching such modeling tasks, and in recent years there has been an explosion of new hierarchical Bayesian models and applications. However, efficient and scalable inference in Bayesian models is a fundamental challenge. In the context of large datasets and increasingly rich hierarchical models, this inference challenge is of central importance to the Bayesian framework, creating a demand both for models that admit powerful inference algorithms and for novel inference strategies.

This thesis addresses some aspects of the Bayesian inference challenge in two parts. In the first part, we study Bayesian models and inference algorithms for time series analysis. We develop new efficient and scalable inference algorithms for Bayesian and Bayesian nonparametric time series models and we discuss new model generalizations which retain both effective algorithms and interpretable structure. In the second part, we study a highly parallelizable variation on the Gibbs sampling inference algorithm. While this algorithm has limited theoretical support and does not provide accurate results or even converge in general, it has achieved considerable empirical success when applied to some Bayesian models of interest, and its easy scalability suggests that it would be of significant practical interest to develop a theoretical understanding for when such algorithms are effective. We develop a theoretical analysis of this sampling inference algorithm in the case of Gaussian models and analyze some of its key properties.

## ■ 1.1 Efficient models and algorithms for Bayesian time series analysis

Bayesian modeling is a natural fit for tasks in unsupervised time series analysis. In such a setting, given time series or other sequential data, one often aims to infer meaningful *states* or *modes* which describe the dynamical behavior in the data, along with statistical patterns that can describe and distinguish those states. Hierarchical Bayesian modeling provides a powerful framework for constructing rich, flexible models based

on this fundamental idea. Such models have been developed for and applied to complex data in many domains, including speech [30, 68, 18] behavior and motion [32, 33, 51], physiological signals [69], single-molecule biophysics [71], brain-machine interfaces [54], handwritten characters [67], and natural language and text [44, 70]. At their core, these models are built on a Bayesian treatment of the ubiquitous Hidden Markov Model (HMM), a model structure which provides both a coherent treatment of the notion of state and temporal dynamics as well as efficient inference algorithms based on dynamic programming and message passing.

A significant advancement for such general-purpose models was the development of the Bayesian nonparametric hierarchical Dirichlet process hidden Markov model (HDP-HMM) [106]. The HDP-HMM allows the complexity of the model to be learned flexibly from the data, and indeed allows the complexity of the representation to grow with the amount of data. However, in the case of the HDP-HMM the flexibility of the Bayesian nonparametric prior can lead to models with undesirably rapid switching dynamics, to the detriment of model interpretability and parsimony. The first work to address this issue was the development of the Sticky HDP-HMM, which introduced a global bias to encourage state persistence [31, 30]. In Johnson and Willsky [60] we generalized the Sticky HDP-HMM to the hierarchical Dirichlet process hidden semi-Markov model (HDP-HSMM), integrating work on HSMMs to allow arbitrary state-specific duration distributions in the Bayesian nonparametric setting. However, as with other HSMM models, explicit duration modeling increases the computational cost of inference, scaling quadratically with the observation sequence length while the cost of HMM inference scales only linearly. This increased computational complexity can limit the applicability of both the HDP-HSMM and the Bayesian HSMM even when non-geometric duration distributions provide a better model.

An additional challenge for all such Bayesian time series models is scaling inference to large datasets. In particular, the Gibbs sampling and mean field algorithms developed for the HMM and HSMM, as well as their nonparametric extensions, require a complete pass over the dataset in each iteration and thus do not scale well. In contrast, recent advances in scalable mean field methods require only a small number of passes over large datasets, often producing model fits in just a single pass. However, while such methods have been studied extensively for text topic models [53, 115, 17, 114, 92, 52], they have not been developed for or applied to time series models.

In this thesis we address these inference challenges in several ways. To address the challenge of expensive HSMM inference, in Chapter 4 we develop a framework for computing HSMM messages efficiently for some duration distributions. In particular, we derive message passing recursions for HSMMs with durations that are modeled as negative binomials or mixtures of negative binomials for which the computational

complexity scales only linearly with the observation sequence length. We also give an HSMM Gibbs sampling algorithm which exploits this efficient structure.

To address the challenge of scaling inference to large datasets, in Chapter 5 we develop stochastic variational inference (SVI) algorithms for the Bayesian HMM, HSMM, and their nonparametric extensions. In addition, we build on the framework of Chapter 4 to develop fast approximate updates for HSMMs with negative binomial durations. We show that, as with SVI for topic models, the resulting algorithms provide speedups of several orders of magnitude for large datasets.

In Chapter 6 we build on these ideas to develop a Bayesian nonparametric time series model that can capture dynamics at multiple timescales while maintaining efficient and scalable inference. This model is applicable to settings in which the states of complex dynamical behavior can be decomposed further into substates with their own dynamics. Such behavior arises naturally in the context of speech analysis, where we may wish to model dynamics within individual phonemes as well as the dynamics across phonemes [68, 18], and in the context of behavior analysis, where complex movements can be decomposed into component parts [51, 32]. While this model is significantly more complex than the HDP-HMM or HDP-HSMM alone, we show how to compose the ideas developed in Chapters 3, 4, and 5 to develop both efficient Gibbs sampling and scalable SVI inference.

## ■ 1.2 Analyzing Hogwild Gaussian Gibbs sampling

Taking a broader perspective on scaling Bayesian inference, it is clear that some of the workhorse Bayesian inference algorithms cannot scale to large datasets for general models. Sampling methods in particular have proven to be hard to scale, and while there is considerable ongoing work on scalable sampling inference [116, 42], new strategies must be developed and analyzed.

Some lines of work aim to parallelize the computation of sampling updates by exploiting conditional independence and graphical model structure [42]. Though this strategy can be effective for some settings, there are many models and collapsed samplers in which there are no exact conditional independencies and hence no opportunities for such parallelization. However, in some cases dependence may be weak enough so that some degree of independent computation can be tolerated, at least to produce good approximate updates. In particular, thorough empirical work has shown that an extremely simple and highly parallelizable strategy can be effective, at least for one one popular hierarchical Bayesian model: by simply running Gibbs updates in parallel on multiple processors and only communicating those updates to other processors periodically, one can effectively sample from the latent Dirichlet allocation (LDA) model [83, 82, 73, 7, 55].

While the empirical success in the case of LDA is well-documented, there is little theory to support this strategy. As a result, it is unclear for which other models or datasets this strategy may be effective, or how the organization of the computation, such as the frequency of synchronization and the number of parallel processors, may affect the results. However, developing a general theory may be difficult: even standard sampling algorithms for general Bayesian models are notoriously resistant to theoretical analysis, and the parallel dynamics add another layer of difficulty.

Therefore to begin to develop such a theoretical analysis we consider the case of using this parallel strategy to sample from Gaussian distributions. Gaussian distributions and algorithms are tractable for analysis because of their deep connection with linear algebra, and we exploit this connection to develop an analysis framework that provides an understanding of several key properties of the algorithm. We call this strategy Hogwild Gaussian Gibbs sampling, and in Chapter 7 we describe our analysis framework and prove several salient results concerning the convergence and correctness of Gaussian Hogwild Gibbs sampling.

## ■ 1.3 Organization and summary of contributions

In this section we provide an outline of the rest of the thesis and a summary of our main contributions.

### Chapter 2: Background

In Chapter 2 we provide a brief overview of the foundations for the work in this thesis, including probabilistic graphical models, exponential family distributions and conjugate Bayesian analysis, hidden Markov models, and Bayesian nonparametric models constructed using the Dirichlet process.

### Chapter 3: The Hierarchical Dirichlet Process Hidden semi-Markov Model

We originally developed the HDP-HSMM in Johnson [59], and we include its development here because Chapters 4, 5, and 6 build on it. There are new contributions in this chapter as well; in particular, in Chapter 3 and Appendix B we provide a thorough numerical study of the Gibbs sampler we proposed for the HDP-HSMM. The power disaggregation application is also new, in addition to the factorial HDP-HSMM model. Finally, the derivations of the Gibbs sampling algorithms are significantly improved.

### Chapter 4: Faster HSMM Inference with Efficient Representations

HSMM message passing is much more computationally expensive than HMM message passing, scaling as $\mathcal{O}(T^2N + TN^2)$ compared to just $\mathcal{O}(TN^2)$ for a model with $N$ states

and an observation sequence of length $T$. This computational cost can severely limit the applicability of models based on the HSMM, especially for larger datasets.

In Chapter 4 we develop a general framework for computing HSMM messages efficiently for specific duration models, so that the computational complexity scales only linearly with $T$. The main practical result, which we use extensively in Chapters 5 and 6, is an efficient message passing recursion for HSMMs with durations that are modeled as negative binomial distributions or mixtures of negative binomial distributions. We also develop a Gibbs sampling algorithm for HSMMs with negative binomial durations using these ideas.

The framework we develop is much more general, and includes new connections to the theory of linear time-invariant (LTI) systems, which could yield efficient message passing recursions for more duration models as well as new approximation schemes.

### Chapter 5: SVI for HMMs, HSMMs, and Nonparametric Extensions

Since scalable inference is a fundamental challenge for Bayesian time series models, in Chapter 5 we apply the stochastic variational inference (SVI) framework to develop new algorithms for Bayesian HMMs, HSMMs, and their nonparametric extensions, the HDP-HMM and HDP-HSMM. We show that these SVI algorithms can fit such time series models with just a single pass over large datasets.

In addition, because the computational complexity of general HSMM message passing inference can be limiting even in the minibatch setting of SVI, we build on the ideas developed in Chapter 4 to develop an approximate SVI update for models with durations that are modeled as negative binomial distributions or mixtures of negative binomial distributions. We demonstrate that this approximate update can effectively fit HSMM models with time complexity that scales only linearly with the observation sequence length.

### Chapter 6: Scalable Inference in Models with Multiple Timescales

In many settings we may wish to learn time series models that represent dynamics at multiple time scales, such as speech models which capture both the dynamics within individual phonemes and the dynamics across phonemes. It is crucial for such models to admit efficient and scalable inference algorithms, since such complex dynamics require larger datasets to be learned effectively.

In Chapter 6 we develop a Bayesian nonparametric model for learning such dynamics. We build on the HDP-HSMM developed in Chapter 3 so that explicit duration modeling can be used to identify dynamics at the different timescales. Using the ideas from Chapters 4 and 5 we also develop an efficient Gibbs sampler and a scalable SVI algorithm. We demonstrate the effectiveness of both the model and the proposed infer-

ence algorithms on an application to unsupervised phoneme discovery.

### Chapter 7: Analyzing Hogwild Parallel Gaussian Gibbs Sampling

Scaling sampling inference algorithms to large datasets is a fundamental challenge for Bayesian inference, and new algorithms and strategies are required. One highly parallelizable strategy, which we call Hogwild Gibbs sampling, has been shown through empirical work to be very effective for sampling from LDA models. However, there is limited theoretical analysis for Hogwild Gibbs sampling and so it is difficult to understand for which models it may be helpful or how algorithm parameters may affect convergence or correctness. Furthermore, sampling algorithms for general models are notoriously difficult to analyze.

In Chapter 7 we develop a theoretical framework for Hogwild Gibbs sampling applied to Gaussian models. By leveraging the Gaussian's deep connection to linear algebra, we are able to understand several properties of the Hogwild Gibbs algorithm, and our framework provides simple linear algebraic proofs. In particular, we give sufficient conditions on the Gaussian precision matrix for the Hogwild Gibbs algorithm to be stable and have the correct process means, we provide an analysis of the accuracy of the process covariances in a low-order analysis regime when cross-processor interactions are small, and we provide a detailed understanding of convergence and accuracy of the process covariance, as well as a way to produce unbiased estimates of the exact covariance, when the number of processor-local Gibbs iterations is large.

### Chapter 8: Conclusions and Recommendations

In Chapter 8 we provide concluding remarks as well as some potential avenues for future research.

Chapter 2

# Background

In this chapter we provide a brief overview of the foundations on which this thesis builds, particularly probabilistic graphical models, exponential family distributions, hidden Markov models, and Bayesian nonparametric models constructed using the Dirichlet process.

## 2.1 Graphical models

In this section we overview the key definitions and results for directed and undirected probabilistic graphical models, which we use both for defining models and constructing algorithms in this thesis. For a more thorough treatment of probabilistic graphical models, see Koller and Friedman [65].

### 2.1.1 Directed graphical models

Directed graphical models, also called Bayes nets, naturally encode generative model parameterizations, where a model is specified via a sequence of conditional distributions. They are particularly useful for the hierarchical Bayesian models and algorithms developed in this thesis.

First, we give a definition of directed graphs and a notion of directed separation of nodes. Next, we connect these definitions to conditional independence structure for collections of random variables and factorization of joint densities.

**Definition 2.1.1** (Directed graph). *For some $n \in \mathbb{N}$, a* directed graph *on $n$ nodes is a pair $(V, E)$ where $V = [n] \triangleq \{1, 2, \ldots, n\}$ and $E \subseteq (V \times V) \setminus \{(i, i) : i \in V\}$. We call the elements of $V$ the (labeled)* nodes *or vertices and the elements of $E$ the* edges, *and we say $(i, j) \in E$ is an edge from $i$ to $j$.*

Given a graph $(V, E)$, for distinct $i, j \in V$ we write $i \to j$ or $j \leftarrow i$ if $(i, j) \in E$ and write $i \quad j$ if $(i, j) \in E$ or $(j, i) \in E$. We say there is a *directed path* from $i_1$ to $i_n$ of length $n - 1$ if for some $i_2, i_3, \ldots, i_{n-1} \in V$ we have $i_1 \to i_2 \to \cdots \to i_n$, and an *undirected path* if we have $i_1 - i_2 - \cdots - i_n$. We say node $j$ is a *descendant* of node $i$

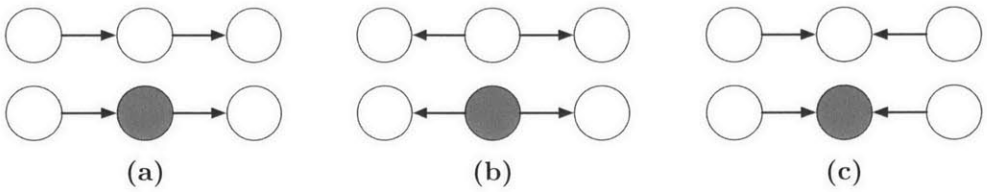



**Figure 2.1:** An illustration of the cases in Definition 2.1.2. Shaded nodes are in the set $C$.

if there is a directed path from $i$ to $j$, and we say $j$ is a *child* of $i$ if there is a directed path of length 1. Similarly, we say $i$ is an *ancestor* of $j$ if $j$ is a descendant of $i$, and we say $i$ is a *parent* of $j$ if $j$ is a child of $i$. We use $\pi_G(i)$ to denote the set of parents of node $i$ and $c_G(i)$ to denote its children.

We say a directed graph is *cyclic* if there is a node $i \in V$ that is its own ancestor, and we say a graph is *acyclic* if it is not cyclic. For directed graphical models, and all of the directed graphs in this thesis, we use *directed acyclic graphs* (DAGs).

Using these notions we can define the main idea of *directed separation.*

**Definition 2.1.2** (Blocked/unblocked triples). *Given a DAG $G = (V, E)$, let $a - b - c$ be an undirected path with $a, b, c \in V$, and let $C \subset V$ be a subset of nodes with $C \cap \{a, c\} = \varnothing$. We call $a - b - c$ a* triple, *and we say it is* blocked by $C$ *in two cases:*

1. *if the structure is not $a \to b \leftarrow c$, then $b \in C$*
2. *if the structure is $a \to b \leftarrow c$, and for all descendants $b' \in V$ of $b$ we have $b' \notin C$.*

*We say a triple is* unblocked by $C$ *if it is not blocked by $C$.*

We illustrate the cases in Definition 2.1.2 in Figure 2.1, which shows six triples of nodes, where nodes in the set $C$ are shaded. In each of (a) and (b), the top triple is unblocked while the bottom triple is blocked, corresponding to case 1 in the definition. However, in (c) the reverse is true: the top triple is blocked while the bottom triple is unblocked, corresponding to case 2 in the definition.

**Definition 2.1.3** (Blocked/unblocked path). *Given a DAG $G = (V, E)$ and a set $C \subset V$, let $i_1 - i_2 - \cdots - i_n$ be a path with $C \cap \{i_1, i_n\} = \varnothing$. We call the path* unblocked by $C$ *if every triple in the path is unblocked by $C$. We call the path* blocked by $C$ *if it is not unblocked.*

Note that Koller and Friedman [65] uses the term *active trail* for our definition of unblocked path.

**Definition 2.1.4** (d-separation). *Given a DAG $G = (V, E)$, for distinct $i, j \in V$ and a subset $C \subset V$ with $C \cap \{i, j\} = \varnothing$, we say $i$ and $j$ are* d-separated in $G$ by $C$

*if there is no undirected path between $i$ and $j$ that is unblocked by $C$, and we write* $\text{d-sep}_G(i, j|C)$. *Further, for disjoint subsets $A, B, C \subset V$ with $A$ and $B$ nonempty we write* $\text{d-sep}_G(A, B|C)$ *if we have* $\text{d-sep}_G(i, j|C)$ *for all $i \in A$ and $j \in B$.*

In words, $i, j \in V$ may not be d-separated in $G$ given $C$ if there exists an undirected path between $i$ and $j$ in $G$. However, the path must be unblocked, where if a node on the path belongs to $C$ it generally blocks the path except when there is a "V" structure $a \to b \leftarrow c$ on the path, in which case $b$ blocks the path unless it or one of its descendants is in $C$. This special rule is useful when defining probabilistic structure in terms of the graph because it models how independent random variables can become dependent when they are competing explanations for the same observation.

Next, we give a definition of conditional independence structure in collections of random variables that uses graphical d-separation.

**Definition 2.1.5** (Markovianity on directed graphs). *Given a DAG $G = (V, E)$ and a collection of random variables $X = \{X_i : i \in V\}$ indexed by labeled nodes in the graph, we say $X$ is* Markov on $G$ *if for disjoint subsets $A, B, C \subset V$ we have*

$$\text{d-sep}_G(A, B|C) \implies X_A \perp\!\!\!\perp X_B | X_C \tag{2.1.1}$$

*where for $S \subseteq V$ we define $X_S \triangleq \{X_i : i \in S\}$.*

Note that this definition does not require that the graph capture all of the conditional independencies present in the collection of random variables. Indeed, a collection of random variables can be Markov on many distinct graphs, and every collection is Markov on the complete graph. Graphs that capture more structure in the collection of random variables are generally more useful.

Conditional independence structure can be used in designing inference algorithms, and a graphical representation can make clear the appropriate notion of local information when designing an algorithm with local updates. A particularly useful notion of local information is captured by the *Markov blanket*.

**Definition 2.1.6** (Directed Markov blanket). *Given a DAG $G = (V, E)$, the* Markov blanket *for node $i \in V$, denoted* $\text{MB}_G(i)$, *is the set of its parents, children, and childrens' parents:*

$$\text{MB}_G(i) \triangleq \{j \in V : j \to i\} \cup \{j \in V : i \to j\} \cup \{j \in V : \exists k \,.\, i \to k \leftarrow j\}. \tag{2.1.2}$$

*The Markov blanket for a set of nodes $A \subseteq V$ contains the Markov blankets for all nodes in $A$ except the nodes in $A$ itself:*

$$\text{MB}_G(A) \triangleq \bigcup_{i \in A} \text{MB}_G(i) \setminus A. \tag{2.1.3}$$

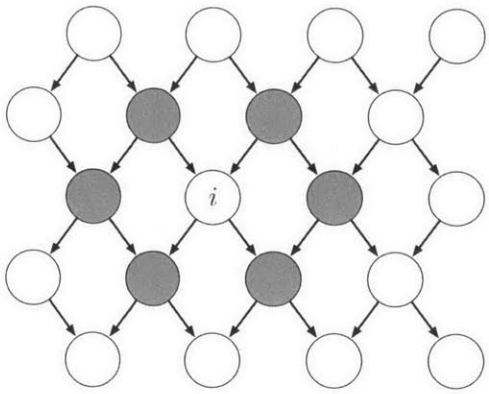



**Figure 2.2:** An illustration of the directed Markov blanket defined in Definition 2.1.6. Nodes in the Markov blanket of node $i$ are shaded gray.

We illustrate Definition 2.1.6 in Figure 2.2. The nodes in the Markov blanket of node $i$ are shaded gray.

**Proposition 2.1.1.** *Given a collection of random variables $\{X_i : i \in V\}$ that is Markov with respect to a DAG $G = (V, E)$, we have*

$$X_i \perp\!\!\!\perp X_S | X_{\mathrm{MB}_G(i)} \tag{2.1.4}$$

*where $S \triangleq V \setminus (\mathrm{MB}_G(i) \cup \{i\})$.*

*Proof.* By conditioning on the parents of node $i$, all paths of the form $a \to b \to i$ are blocked. By conditioning on its children, all paths of the form $i \to b \to c$ are blocked. By conditioning on the childrens' parents, all paths of the form $i \to b \leftarrow c$, which may have been unblocked by conditioning on $b$ or one of its descendants via case 2 of Definition 2.1.4, are blocked. □

Another common and convenient notion of probabilistic graphical structure is a density's factorization with respect to a DAG.

**Definition 2.1.7** (Factoring on directed graphs)**.** *Given a DAG $G = (V, E)$ on $n$ nodes and a collection of random variables $X = \{X_i : i \in V\}$ with density $p_X$ with respect to some base measure, we say $p_X$* factorizes according to *$G$ if we can write*

$$p_X(x_1, \ldots, x_n) = \prod_{i \in V} p(x_i | x_{\pi_G(i)}) \tag{2.1.5}$$

*where $\pi_G(i) \triangleq \{j \in V : j \to i\}$ denotes the set of parents of node $i$ in $G$.*

**Theorem 2.1.1.** *A collection of random variables $\{X_i : i \in V\}$ with a joint density (with respect to some base measure) is Markov on a DAG $G$ if and only if the joint density factorizes as in Eq. (2.1.5).*

*Proof.* The proof is straightforward. In Koller and Friedman [65], Theorem 3.1 shows Markovianity implies the densities factor and Theorem 3.2 shows the reverse. □

## ■ 2.1.2 Undirected graphical models

Undirected graphical models, also called Markov random fields, do not easily encode generative model specifications. However, they can be more useful for encoding soft constraints or local partial correlations. We use an undirected graphical model perspective in our analysis of Hogwild Gibbs Sampling in Chapter 7.

As with directed graphical models, we first define undirected graphs and a notion of separation of nodes, then give definitions that link the graphical structure to both conditional independence structure in a collection of random variables and factorization structure in the joint density for those variables.

**Definition 2.1.8** (Undirected graph). *For some $n \in \mathbb{N}$, an* undirected graph *on $n$ nodes is a pair $(V, E)$ where $V = [n]$ and $E \subseteq \{\{i, j\} : i, j \in V,\ i \neq j\}$.*

Analogous to the definition in the previous section, there is a natural notion of an undirected path between nodes. Given a graph $(V, E)$, for distinct $i, j \in V$ we write $i - j$ if $\{i, j\} \in E$, and we say there is an (undirected) *path* from $i_1$ to $i_n$ of length $n-1$ if for some $i_2, i_3, \ldots, i_{n-1} \in V$ we have $i_1 - i_2 - \cdots - i_n$. We say $i$ is a *neighbor* of $j$ if $\{i, j\} \in E$ and denote the set of neighbors of node $i$ as $n_G(i) \triangleq \{j \in V : \{i, j\} \in E\}$. We say a pair of nodes is *connected* if there exists an (undirected) path from $i$ to $j$.

The notion of undirected separation and corresponding notion of Markovianity on undirected graphs is simpler than those for directed graphs.

**Definition 2.1.9** (Undirected separation). *Given an undirected graph $G = (V, E)$, for distinct $i, j \in V$ and a subset $C \subset V$ with $C \cap \{i, j\} = \varnothing$, we say $i$ and $j$ are* separated *in $G$ given $C$ and write $\mathrm{sep}_G(i, j|C)$ if there is no path from $i$ to $j$ that avoids $C$. For disjoint subsets $A, B, C \subset V$ with $A$ and $B$ nonempty we write $\mathrm{sep}_G(A, B|C)$ if we have $\mathrm{sep}_G(i, j|C)$ for all $i \in A$ and $j \in B$.*

**Definition 2.1.10** (Markovianity on undirected graphs). *Given an undirected graph $G = (V, E)$ and a collection of random variables $X = \{X_i : i \in V\}$ indexed by labeled nodes in the graph, we say $X$ is* Markov on *$G$ if for disjoint subsets $A, B, C \subset V$ we have*

$$\mathrm{sep}_G(A, B|C) \implies X_A \perp\!\!\!\perp X_B | X_C. \tag{2.1.6}$$

As with Markovianity with respect to directed graphs, an undirected graph may not encode all the conditional independence statements possible for a given collection of random variables.

The notion of Markov blanket for an undirected graph is also simpler.

**Definition 2.1.11** (Undirected Markov blanket). *Given an undirected graph $G = (V, E)$, the* Markov blanket *for each node $i \in V$, denoted $\mathrm{MB}_G(i)$, is the set of neighbors of node $i$:*

$$\mathrm{MB}_G(i) \triangleq n_G(i) = \{j \in V : \{i, j\} \in E\}. \tag{2.1.7}$$

*The Markov blanket for a set of nodes $A \subseteq V$ is the set of all neighbors to nodes in $A$ excluding those in $A$:*

$$\mathrm{MB}_G(A) \triangleq \bigcup_{i \in A} \mathrm{MB}_G(i) \setminus A. \tag{2.1.8}$$

We can use this definition for an undirected analog of Proposition 2.1.1.

**Proposition 2.1.2.** *Given a collection of random variables $\{X_i : i \in V\}$ that is Markov with respect to an undirected graph $G = (V, E)$, we have*

$$X_i \perp\!\!\!\perp X_S | X_{\mathrm{MB}_G(i)} \tag{2.1.9}$$

*where $S \triangleq V \setminus (\mathrm{MB}_G(i) \cup \{i\})$.*

*Proof.* Because all the neighbors of $i$ are in $\mathrm{MB}_G(i)$, for any $j \in S$ there can be no undirected path from $j$ to $i$ that avoids $\mathrm{MB}_G(i)$. □

We can also define a density factorization with respect to an undirected graph, though we must first define the notion of a clique. A *clique* in a graph $(V, E)$ is a nonempty subset of fully-connected nodes; that is, a nonempty set $C \subseteq V$ is a clique if for every distinct $i, j \in C$ we have $\{i, j\} \in E$.

**Definition 2.1.12** (Factoring on undirected graphs). *Given an undirected graph $G = (V, E)$ and a collection of random variables $X = \{X_i : i \in V\}$ with density $p_X$ with respect to some base measure, we say $p_X$* factorizes according to *$G$ if we can write*

$$p_X(x_1, \ldots, x_n) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(x_C) \tag{2.1.10}$$

*for a collection of cliques $\mathcal{C}$ of $G$ and nonnegative* potentials *or* factors *$\{\psi_C : C \in \mathcal{C}\}$ indexed by those cliques, where*

$$Z \triangleq \int \prod_{C \in \mathcal{C}} \psi_C(x_C) \nu(dx) \tag{2.1.11}$$

*is a normalizing constant.*

Note that cliques typically overlap; that is, we may have $C_i \cap C_j \neq \varnothing$ for distinct $C_i, C_j \in \mathcal{C}$. To remove many possible redundancies, without loss of generality one can assume that $\mathcal{C}$ includes only *maximal cliques*, where a maximal clique cannot have any other node added to it and remain a (fully-connected) clique.

The correspondence between a collection of random variables being Markov on an undirected graph and its joint density factorizing as Eq. (2.1.10) is not quite as simple as that for directed graphs because deterministic relationships among the random variables can prevent factoring the density, as shown in the next example.

**Example 2.1.1.** *Using Example 4.4 from Koller and Friedman [65], consider four binary random variables* $X = \{X_i : i = 1, 2, 3, 4\}$ *with a PMF that takes value* $1/8$ *on the configurations of* $(x_1, x_2, x_3, x_4)$ *given by*

$$\begin{array}{cccc} (0,0,0,0) & (1,0,0,0) & (1,1,0,0) & (1,1,1,0) \\ (0,0,0,1) & (0,0,1,1) & (0,1,1,1) & (1,1,1,1) \end{array}$$

*and* $0$ *elsewhere. Then* $X$ *is Markov on the graph* $1 - 2 - 3 - 4 - 1$ *but the density cannot be factorized into pairwise potentials.*

This issue cannot arise when we restrict our attention to strictly positive densities.

**Theorem 2.1.2.** *Given a collection of random variables* $X = \{X_i : i \in V\}$ *with a joint density* $p_X$ *(with respect to some base measure) and an undirected graph* $G$*, we have*

1. *If* $p_X$ *factorizes according to* $G$*, then* $X$ *is Markov on* $G$*.*
2. *If* $X$ *is Markov on* $G$ *and* $p_X$ *is strictly positive, then* $p_X$ *factors according to* $G$*.*

*Proof.* The proof for 1 is straightforward and is given as the proof of Theorem 4.1 in Koller and Friedman [65]. The proof for 2 is the proof of the Hammersley-Clifford theorem, given as Theorem 4.8 in Koller and Friedman [65]. □

### ■ 2.1.3 Exact Inference and Graph Structure

Given some specification of a probability distribution, *inference* for that distribution means computing quantities of interest such as marginals, conditionals, or expectations. In the graphical model framework we can be precise about the computations required to perform inference and their complexity, as we overview in this subsection. For concreteness, we focus on undirected models with densities.

Consider an undirected graphical model specified by an undirected graph $G = (V, E)$ and a set of potentials $\{\psi_C : C \in \mathcal{C}\}$ on a set $\mathcal{C}$ of cliques of $G$. The joint density is

proportional to the product of the clique potentials $p(x) \propto \prod_{C \in \mathcal{C}} \psi_C(x_C)$. We can represent an arbitrary inference query in terms of a subset of nodes on which we condition, a subset of nodes which we marginalize out, and a subset over which we want to represent the resulting density; that is, we partition the set of nodes $V$ into three disjoint (possibly empty) subsets $A_1$, $A_2$, $A_3$ with $A_1 \cup A_2 \cup A_3 = V$ and write

$$p(x_{A_1}|x_{A_3}) = \int p(x_{A_1}, x_{A_2}|x_{A_3})\nu(dx_{A_2}) = \frac{\int \prod_{C \in \mathcal{C}} \psi_C(x_C)\nu(dx_{A_2})}{\int \prod_{C \in \mathcal{C}} \psi_C(x_C)\nu(dx_{A_1 \cup A_2})}. \tag{2.1.12}$$

Therefore to compute an arbitrary inference query we need only to compute integrals of products of the factors in the density. To simplify notation, we write such computations in the form

$$\int \prod_{C \in \mathcal{C}} \psi_C(x_C)\nu(dx_A) \tag{2.1.13}$$

for some subset $A \subseteq V$.

Graph structure affects how we can organize a computation of the form (2.1.13) and thus its computational complexity. Consider the special case of integrating out a single variable $x_j$ by partitioning the set of cliques into those which contain node $j$ and those which do not, $\mathcal{C}_j \triangleq \{C \in \mathcal{C} : j \in C\}$ and $\mathcal{C}_{\setminus j} \triangleq \{C \in \mathcal{C} : j \notin C\}$, and writing

$$\int \prod_{C \in \mathcal{C}} \psi_C(x_C)\nu(dx_j) = \prod_{C_{\setminus j} \in \mathcal{C}_{\setminus j}} \psi_{C_{\setminus j}}(x_{C_{\setminus j}}) \int \prod_{C_j \in \mathcal{C}_j} \psi_{C_j}(x_{C_j})\nu(dx_j) \tag{2.1.14}$$

$$= \prod_{C_{\setminus j} \in \mathcal{C}_{\setminus j}} \psi_{C_{\setminus j}}(x_{C_{\setminus j}})\psi_B(x_B) \tag{2.1.15}$$

where $B \triangleq \{i \in C_j : C_j \in \mathcal{C}_j\} \setminus \{j\}$ is the set of all indices that share a clique with node $j$ in $G$ and $\psi_B$ is a new factor on the clique $B$ resulting from the integral over $x_j$. Thus as a result of the integral there is an *induced graph* on $V \setminus \{j\}$ formed by *eliminating* $j$ by fully connecting its neighbors and deleting it from the graph.

When integrating over multiple variables, the process repeats: given an *elimination order*, nodes are eliminated one by one from the graph, and each elimination introduces a new clique in the graph and a corresponding new potential term in the density over the remaining variables. The computational complexity of the process is determined by the size of the largest clique encountered. In the case of PMFs with finite support, the number of entries in the table encoding the new potential formed in (2.1.15) is typically exponential in the size of the new clique; in the case of PDFs, the complexity depends on the complexity of integrating over factors on cliques, where the clique size determines the number of dimensions in the domain of the integrand, and the complexity

of representing the result. The optimal elimination order is itself NP hard to find for general graphs.

Some classes of graphs have straightforward elimination orderings that avoid the growth in clique size, and inference in distributions that are Markov on such graphs avoids the corresponding growth in complexity.

**Definition 2.1.13** (Tree graph). *A directed or undirected graph $G = (V, E)$ is a* tree *if for each pair distinct nodes $i, j \in V$ there is a unique undirected path from $i$ to $j$ in $G$.*

In an undirected tree $G$, we refer to the nodes $i$ with only single neighbors $|n_G(i)| = 1$ as *leaves*. In a directed tree, we refer to the nodes with no children as leaves.

**Proposition 2.1.3** (Markov on trees). *A density $p$ that factors on an undirected tree $(V, E)$ can be written in terms of pairwise factors $\{\psi_{ij} : \{i, j\} \in E\}$ as*

$$p(x) = \frac{1}{Z} \prod_{\{i,j\} \in E} \psi_{ij}(x_i, x_j). \tag{2.1.16}$$

*Proof.* All of the maximal cliques in an undirected tree are of size at most 2. □

Note that because the edges are undirected, $\psi_{ij}$ and $\psi_{ji}$ denote the same object.

Given a density that factors according to an undirected tree specified in terms of its pairwise factors $\{\psi_{ij} : \{i, j\} \in E\}$ we can convert it to a directed specification by local normalization, i.e. by choosing a direction for each edge and computing

$$p(x_i | x_j) = \frac{\psi_{ij}(x_i, x_j)}{\int \psi_{ij}(x_i, x_{j'}) \nu(dx_{j'})}. \tag{2.1.17}$$

Note that a density that factors on a directed tree may not in general be written purely in terms of conditional probability factors that depend only on pairs of nodes, as shown in the next example.

**Example 2.1.2.** *A density that factors according to the directed tree $G = (V, E)$ with $V = \{1, 2, 3\}$ and edges $1 \to 2 \leftarrow 3$ may include a factor $p(x_2 | x_1, x_3)$. Such a density is not Markov on the undirected tree $G' = (V', E')$ with $V' = V$ and edges $1 - 2 - 3$, and instead only factors on the complete undirected graph with edges $1 - 2 - 3$ 1.*

Directed trees in which each node has at most one parent avoid this issue, and we can convert freely between directed and undirected tree parameterizations for such models.

**Proposition 2.1.4.** *A density $p$ that factors on a directed tree $G = (V, E)$ in which each node only has one parent, i.e. $|\pi_G(i)| \leq 1$ for $i \in V$, also factors with respect to*

*the undirected tree $G' = (V', E')$ formed by dropping the directions on the edges of $G$, i.e. $V' = V$ and $E' = \{\{i, j\} : (i, j) \in E\}$*

*Proof.* Set $\psi_{ij}(x_i, x_j) = p(x_i|x_j)$. □

With undirected trees and directed trees in which each node has at most one parent we can perform elimination without introducing larger factors by using an elimination order that recursively eliminates leaves. Furthermore, the corresponding partial sums for all such elimination orderings can be computed simultaneously with an efficient dynamic programming algorithm, as shown in the next theorem.

For an undirected graph $G = (V, E)$ and a subset of nodes $A \subseteq V$, we define $G \setminus A$ to be the graph formed by deleting the nodes $A$ and the edges incident on nodes in $A$, i.e. the graph $(V', E')$ where $V' = V \setminus A$ and $E' = \{\{i, j\} \in E : i, j \notin A\}$. We say a pair of nodes is *connected* if there exists an undirected path from $i$ to $j$, and we say subsets $A, B \subset V$ are connected if there exists an $i \in A$ and $j \in B$ that are connected.

**Definition 2.1.14** (Tree messages). *Given a density with respect to a base measure $\nu$ that factorizes on an undirected tree $G = (V, E)$ of the form (2.1.16), we define the* message from node $i$ to node $j$ *with $\{i, j\} \in E$ to be*

$$m_{j \to i}(x_i) \triangleq \int \prod_{\{i', j'\} \in E'} \psi_{i'j'}(x_{i'}, x_{j'}) \psi_{ij}(x_i, x_j) \nu(dx_{V'}) \tag{2.1.18}$$

*where $(V', E')$ is the subtree of $G$ that is disconnected from node $i$ in $G \setminus \{j\}$.*

**Theorem 2.1.3** (Tree message passing). *For a density that factorizes on an undirected tree $G = (V, E)$ of the form (2.1.16), the result of any computation of the form (2.1.13) can be written in terms of messages as*

$$\int \prod_{\{i,j\} \in E} \psi_{ij}(x_i, x_j) \nu(dx_A) = \prod_{\{i,j\} \in E'} \psi_{ij}(x_i, x_j) \prod_{k \in B} m_{j \to k}(x_k) \tag{2.1.19}$$

*where $(V', E') = G \setminus A$ is the graph over the the nodes that are not integrated out and $B$ is the set of nodes in $V'$ that have edges to $A$ in $G$, i.e. $B = \{k \in V' : \{k, j\} \in E, j \in A\}$. Furthermore, all messages can be computed efficiently and simultaneously via the recursions*

$$m_{i \to j}(x_i) = \int \psi_{ij}(x_i, x_j) \prod_{k \in n_G(j) \setminus \{i\}} m_{j \to k}(x_j) \nu(dx_j). \tag{2.1.20}$$

*Therefore all inference queries in undirected trees can be computed in time linear in the length of the longest path in the tree.*

*Proof.* The theorem follows from applying elimination to trees and expressing every partial elimination result as (2.1.20). □

We refer to an implementation of the recursion (2.1.20) as a *tree message-passing algorithm.* We use tree message passing algorithms extensively as subroutines in the inference algorithms for the time series models that we develop in the sequel.

## ■ 2.2 Exponential families and conjugacy

In this section, we define exponential families and give some of the properties that we use in this thesis.

### ■ 2.2.1 Definition and Basic Properties

**Definition 2.2.1** (Exponential family). *We say a family of densities $p$ with respect to a base measure $\nu$ indexed by a parameter vector $\theta$ is an* exponential family *of densities if it can be written as*

$$p(x|\theta) = h(x)\exp\{\langle \eta(\theta), t(x)\rangle - Z(\eta(\theta))\} \tag{2.2.1}$$

*where $\langle\cdot,\cdot\rangle$ is an inner product on real vector spaces. We call $\eta(\theta)$ the* natural parameter *vector, $t(x)$ the (sufficient)* statistic *vector, $h(x)$ the* base density, *and*

$$Z(\eta) \triangleq \ln \int e^{\langle \eta, t(x)\rangle} h(x)\nu(dx) \tag{2.2.2}$$

*the* log partition function.

It is often useful to parameterize the family directly in terms of $\eta$, in which case we simply write the density as $p(x|\eta)$. Note that marginalizing over part of the support of an exponential family, such as marginalizing over one coordinate of $x$ or of $t(x)$, does not in general yield another exponential family.

Given an exponential family of the form (2.2.1) we define the set of natural parameters that yield valid normalizable probability densities as $\Theta$, where

$$\Theta \triangleq \{\eta : Z(\eta) < \infty\} \tag{2.2.3}$$

and the set of realizable expected statistics as

$$\mathcal{M} \triangleq \{\mathbb{E}_{X\sim p(\cdot|\eta)}[t(X)] : \eta \in \Theta\} \tag{2.2.4}$$

where $X \sim p(\,\cdot\,|\eta)$ denotes that $X$ is distributed with density $p(\,\cdot\,|\eta)$. We say a family is *regular* if $\Theta$ is open, and *minimal* if there is no nonzero $a$ such that $\langle a, t(x)\rangle$ is equal to a constant ($\nu$-a.e.). Minimality ensures that there is a unique natural parameter for each possible density (up to values on sets of $\nu$-measure 0). We say a family is *tractable*

if for any $\eta$ we can evaluate $Z(\eta)$ efficiently[1] and when $X \sim p(\,\cdot\,|\eta)$ we can compute $\mathbb{E}[t(X)]$ and simulate samples of $X$ efficiently.

For a regular exponential family, derivatives of $Z$ are related to expected statistics.

**Proposition 2.2.1** (Mean mapping and cumulants). *For a regular exponential family of densities of the form* (2.2.1) *with* $X \sim p(\,\cdot\,|\eta)$, *we have* $\nabla Z : \Theta \to \mathcal{M}$ *and*

$$\nabla Z(\eta) = \mathbb{E}[t(X)] \tag{2.2.5}$$

*and writing* $\mu \triangleq \mathbb{E}[t(X)]$ *we have*

$$\nabla^2 Z(\eta) = \mathbb{E}[t(X)t(X)^\mathsf{T}] - \mu\mu^\mathsf{T}. \tag{2.2.6}$$

*More generally, the moment generating function can be written*

$$\mathrm{M}_X(s) \triangleq \mathbb{E}[e^{\langle s, X\rangle}] = e^{Z(\eta+s)-Z(\eta)}. \tag{2.2.7}$$

*and so derivatives of* $Z$ *generate* cumulants *of* $X$, *where the first cumulant is the mean and the second and third cumulants are the second and third central moments, respectively.*

*Proof.* Note that we require the family to be regular even to define $\nabla Z$. To show 2.2.5, using $\nu$ as the base measure for the density we write

$$\begin{aligned}
\nabla_\eta Z(\eta) &= \nabla_\eta \ln \int e^{\langle \eta, t(x)\rangle} h(x)\nu(dx) && (2.2.8)\\
&= \frac{1}{\int e^{\langle \eta, t(x)\rangle} h(x)\nu(dx)} \int t(x) e^{\langle \eta, t(x)\rangle} h(x)\nu(dx) && (2.2.9)\\
&= \int t(x) p(x|\eta)\nu(dx) && (2.2.10)\\
&= \mathbb{E}[t(X)]. && (2.2.11)
\end{aligned}$$

To derive the form of the moment generating function, we write

$$\begin{aligned}
\mathbb{E}[e^{\langle s, X\rangle}] &= \int e^{\langle s, x\rangle} p(x)\nu(dx) && (2.2.12)\\
&= \int e^{\langle s, x\rangle} e^{\langle \eta, t(x)\rangle - Z(\eta)} h(x)\nu(dx) && (2.2.13)\\
&= e^{Z(\eta+s)-Z(\eta)}. && (2.2.14)
\end{aligned}$$

[1] We do not provide a precise definition of computational efficiency here. Common definitions often correspond to the complexity classes P or BPP [3].

The cumulant generating function for $X$ is then $\ln \mathrm{M}_X(s) = Z(\eta + s) - Z(\eta)$. □

When a specific set of expected statistics can only arise from one member of an exponential family, we say the family is *identifiable* and we can use the moments as an alternative way to parameterize the family.

**Theorem 2.2.1** (Exponential family identifiability). *For a regular, minimal exponential family of the form* (2.2.1), $\nabla Z : \Theta \to \mathcal{M}$ *is injective, and* $\nabla Z : \Theta \to \mathcal{M}^\circ$ *is surjective, where* $\mathcal{M}^\circ$ *denotes the interior of* $\mathcal{M}$. *Therefore* $\nabla Z : \Theta \to \mathcal{M}^\circ$ *is a bijection.*

*Proof.* See Wainwright and Jordan [113, Theorem 3.3]. □

When parameterizing a regular, minimal exponential family in terms of expected statistics $\mu \in \mathcal{M}^\circ$, we say it is written with *mean parameters*, and we have $\eta(\mu) = (\nabla Z)^{-1}(\mu)$ using Theorem 2.2.1. Given a set of moments, the corresponding minimal exponential family member has a natural interpretation as the density (relative to $\nu$) with maximum entropy subject to those moment constraints [113, Section 3.1].

For members of an exponential family, many quantities can be expressed generically in terms of the natural parameter, expected statistics under that parameter, and the log partition function.

**Proposition 2.2.2** (Entropy, score, and Fisher information). *For a regular exponential family of densities of the form* (2.2.1) *parameterized in terms of its natural parameter* $\eta$, *with* $X \sim p(\,\cdot\,|\eta)$ *and* $\mu(\eta) \triangleq \mathbb{E}[t(X)]$ *we have*

1. *The (differential)* entropy *is*

$$\mathbb{H}[p] \triangleq -\mathbb{E}[\ln p(X|\eta)] = -\langle \eta, \mu(\eta) \rangle + Z(\eta). \tag{2.2.15}$$

2. *When the family is regular, the* score *with respect to the natural parameter is*

$$v(x, \eta) \triangleq \nabla_\eta \ln p(x|\eta) = t(x) - \mu(\eta) \tag{2.2.16}$$

3. *When the family is regular, the* Fisher information *with respect to the natural parameter is*

$$\mathcal{I}(\eta) \triangleq \mathbb{E}[v(X, \eta) v(X, \eta)^\mathsf{T}] = \nabla^2 Z(\eta). \tag{2.2.17}$$

*Proof.* Each follows from (2.2.1), where 2.2.16 and 2.2.17 use Proposition 2.2.1. □

When the family is parameterized in terms of some other parameter $\theta$ so that $\eta = \eta(\theta)$, the properties in Proposition 2.2.2 include Jacobian terms of the form $\partial\eta/\partial\theta$.

When the alternative parameter is the mean parameter $\mu$, since $\eta(\mu) = (\nabla Z)^{-1}(\mu)$ the relevant Jacobian is $(\nabla^2 Z)^{-1}$. We use the Fisher information expression (2.2.17) in the development of a stochastic variational inference (SVI) algorithm in Chapter 5.

We conclude this subsection with two examples of exponential families of densities.

**Example 2.2.1** (Gaussian). *The Gaussian PDF can be parameterized in terms of its mean and covariance $(\mu, \Sigma)$, where $\mu \in \mathbb{R}^d$ and $\Sigma \in \mathbb{R}^{d\times d}$ with $\Sigma \succ 0$, and can be written*

$$\begin{aligned} p(x|\mu,\Sigma) &= (2\pi)^{-d/2}|\Sigma|^{-1/2}\exp\left\{-\frac{1}{2}(x-\mu)^\mathsf{T}\Sigma^{-1}(x-\mu)\right\} \\ &= \underbrace{(2\pi)^{-d/2}}_{h_\mathrm{N}(x)}\exp\Big\{\big\langle \underbrace{(-\tfrac{1}{2}\Sigma^{-1}, \Sigma^{-1}\mu, -\tfrac{1}{2}\mu^\mathsf{T}\Sigma^{-1}\mu)}_{\eta_\mathrm{N}(\mu,\Sigma)}, \underbrace{(xx^\mathsf{T}, x, 1)}_{t_\mathrm{N}(x)}\big\rangle - \underbrace{\tfrac{1}{2}\ln|\Sigma|}_{Z_\mathrm{N}(\eta_\mathrm{N}(\mu,\Sigma))}\Big\}. \end{aligned}$$

*where $\langle (A,b,c),(D,e,f)\rangle = \mathrm{tr}(A^\mathsf{T}D) + b^\mathsf{T}e + cf$. Therefore it is an exponential family of densities, and further it is a regular exponential family since $\Theta = \{(\Sigma,\mu) : \Sigma \succ 0, \mu \in R^d\}$ is open (in the standard product topology for Euclidean spaces). The family is minimal because there is no nonzero $(A,b,c)$ such that $\langle (A,b,c), t_\mathrm{N}(x)\rangle = x^\mathsf{T}Ax + b^\mathsf{T}x + c$ equal to a constant ($\nu$-a.e.).*

**Example 2.2.2** (Categorical). *Consider drawing a sample $X$ from a finite distribution with support on $[K] = \{1, 2, \ldots, K\}$, where $p(x = k|\pi) = \pi_k$ for $K \in [K]$ and $\pi$ satisfies $\sum_i \pi_i = 1$ and $\pi > 0$ element-wise. We can write the PMF for $X$ as*

$$p(x|\bar{\pi}) = \prod_{k=1}^{K} \pi_k^{\mathbb{I}[x=k]} = \exp\Big\{\sum_{k=1}^{K} \ln \pi_k \mathbb{I}[x=k]\Big\} = \exp\big\{\langle \ln \pi, \mathbb{1}_x\rangle\big\} \tag{2.2.18}$$

*where the log is taken element-wise, $\mathbb{I}[\,\cdot\,]$ is an indicator function that takes value 1 when its argument is true and 0 otherwise, and $\mathbb{1}_k$ is an indicator vector with its $i$th entry $\mathbb{I}[k=i]$. We call this family of densities the* categorical *family, and it is an exponential family of densities with natural parameter $\eta(\pi) = \ln \pi$ and statistic $t(x) = \mathbb{1}_x$. (A closely related family is the* multinomial *family, where we consider drawing a set of $n$ independent samples from the same process, $x = \{x_i : i \in [n]\}$, and defining the statistic to be the counts of each occurrence, i.e. the $k$th entry of $t(x)$ is $|\{x_i : x_i = k\}|$.)*

*Note that $Z(\pi) = 0$. The categorical family as written in (2.2.18) is not a regular exponential family because $\Theta = \{\pi \in \mathbb{R}^K : \sum_i \pi_i = 1,\ \pi > 0\}$ is not open. Since the family is not regular, (2.2.16) does not apply. We can instead write the family of densities as*

$$p(x|\pi) = \exp\big\{\langle \ln \bar{\pi}, \mathbb{1}_x\rangle - \ln \textstyle\sum_{i=1}^K \bar{\pi}_i\big\} \tag{2.2.19}$$

*where $Z(\eta(\bar{\pi})) = \ln \sum_{i=1}^K \bar{\pi}_i$ so that $\Theta = \{\bar{\pi} \in \mathbb{R}^K : \pi > 0\}$ is open. However, neither*

(2.2.18) *or* (2.2.19) *is a minimal exponential family, since the statistic satisfies* $\langle 1, \mathbb{1}_x \rangle = 1$ *for any* $x \in [K]$. *As is clear from the renormalization in* (2.2.19), *the parameter is not identifiable, so Theorem 2.2.1 does not apply.*

*While it is possible to write the categorical as a regular minimal exponential family by removing one component from the parameterization, it is often easiest to work with the categorical (and multinomial) in the form* (2.2.18).

## ■ 2.2.2 Conjugacy

In this subsection, we define a notion of conjugacy for pairs of families of distributions. Conjugate families are especially useful for Bayesian analysis and algorithms.

**Definition 2.2.2.** *Given two (not necessarily exponential) families of densities* $p_1(\theta|\alpha)$ *and* $p_2(x|\theta)$ *indexed by parameters* $\alpha$ *and* $\theta$, *respectively, we say the pair* $(p_1, p_2)$ *are a* conjugate pair of densities *if for all* $\alpha$, $x$, *and* $\theta$ *we have*

$$p_1(\theta|\alpha)p_2(x|\theta) \propto p_1(\theta|\alpha') \tag{2.2.20}$$

*for some* $\alpha' = \alpha'(x, \alpha)$ *that may depend on* $x$ *and* $\alpha$.

*We can extend this definition to distributions without densities by considering instead indexed families of laws [88, Definition 2].*

Conjugate pairs are particularly useful in Bayesian analysis because if we have a *prior family* $p(\theta|\alpha)$ and we observe data generated according to a *likelihood* $p(x|\theta)$ then the *posterior* $p(\theta|x, \alpha)$ is in the same family as the prior. In the context of Bayesian updating, we call $\alpha$ the *hyperparameter* and $\alpha'$ the *posterior hyperparameter.*

Given a regular exponential family likelihood, we can always define a conjugate prior, as shown in the next proposition.

**Proposition 2.2.3.** *Given a regular exponential family*

$$p_{X|\theta}(x|\theta) = h_X(x) \exp\{\langle \eta_X(\theta), t_X(x) \rangle - Z_X(\eta_X(\theta))\} \tag{2.2.21}$$

$$= h_X(x) \exp\{\langle (\eta_X(\theta), -Z_X(\eta(\theta))), (t_X(x), 1) \rangle\} \tag{2.2.22}$$

*then if we define the statistic* $t_\theta(\theta) \triangleq (\eta_X(\theta), -Z_X(\eta(\theta)))$ *and an exponential family of densities with respect to that statistic as*

$$p_{\theta|\alpha}(\theta|\alpha) = h_\theta(\theta) \exp\{\langle \eta_\theta(\alpha), t_\theta(\theta) \rangle - Z_\theta(\eta_\theta(\alpha))\} \tag{2.2.23}$$

*then the pair* $(p_{\theta|\alpha}, p_{X|\theta})$ *is a conjugate pair of families with*

$$p(\theta|\alpha)p(x|\theta) \propto h_\theta(\theta)\exp\{\langle \eta_\theta(\alpha) + (t_X(x), 1), t_\theta(\theta)\rangle\} \tag{2.2.24}$$

*and hence we can write the posterior hyperparameter as* $\alpha' = \eta_\theta^{-1}(\eta_\theta(\alpha) + (t_X(x), 1))$. *When the prior family is parameterized with its natural parameter, we have* $\eta' = \eta + (t_X(x), 1)$.

As a consequence of Proposition 2.2.3, if the prior family is written with natural parameters and we generate data $\{x_i\}_{i=1}^n$ according to the model

$$\theta \sim p_{\theta|\eta}(\,\cdot\,|\eta) \tag{2.2.25}$$

$$x_i|\theta \overset{\text{iid}}{\sim} p_{X|\theta}(\,\cdot\,|\theta) \quad i = 1, 2, \ldots, n, \tag{2.2.26}$$

where the notation $x_i \overset{\text{iid}}{\sim} p(\,\cdot\,)$ denotes that the random variables $x_i$ are independently and identically distributed, then $p(\theta|\{x_i\}_{i=1}^n, \eta)$ has posterior hyperparameter $\eta' = \eta + (\sum_{i=1}^n t(x_i), n)$. Therefore if a prior family $p(\,\cdot\,|\eta)$ is tractable then the posterior under the conjugate likelihood is tractable.

We conclude this subsection with two examples of conjugate pairs of exponential families. A list of conjugate pairs can be found in Gelman et al. [38].

**Example 2.2.3** (NIW-Gaussian conjugacy). *Here we show that the normal-inverse-Wishart (NIW) is a conjugate prior family for the Gaussian likelihood of Example 2.2.1.*

*The NIW density with parameter* $\alpha = (\Lambda_0, \mu_0, \kappa_0, \nu_0)$ *and* $\Lambda_0 \succ 0$, $\kappa_0 > 0$, $\nu_0 > d$ *is*

$$\begin{aligned} p(\mu, \Sigma|\alpha) &\propto |\Sigma|^{-((\nu_0+d)/2+1)} \exp\Big\{-\frac{1}{2}\operatorname{tr}(\Lambda_0\Sigma^{-1}) - \frac{\kappa_0}{2}(\mu-\mu_0)^\mathsf{T}\Sigma^{-1}(\mu-\mu_0)\Big\} \\ &= \underbrace{|\Sigma|^{-d/2+1}}_{h_{\text{NIW}}(\mu,\Sigma)} \exp\Big\{\Big\langle \underbrace{(\Lambda_0 + \kappa_0\mu_0\mu_0^\mathsf{T}, \kappa_0\mu_0, \kappa_0, \nu_0)}_{\eta_{\text{NIW}}(\Lambda_0,\mu_0,\kappa_0,\nu_0)}, \underbrace{(-\tfrac{1}{2}\Sigma^{-1}, \Sigma^{-1}\mu, -\tfrac{1}{2}\mu^\mathsf{T}\Sigma^{-1}\mu, -\tfrac{1}{2}\ln|\Sigma|)}_{t_{\text{NIW}}(\mu,\Sigma)} \Big\rangle\Big\} \end{aligned}$$

*where proportionality is over* $(\mu, \Sigma)$ *and* $\langle (A, b, c, d), (E, f, g, h)\rangle = \operatorname{tr}(A^\mathsf{T}E) + b^\mathsf{T}f + cg + dh$.

*The normal density from Example 2.2.1 is*

$$\begin{aligned} p(x|\mu, \Sigma) &= \underbrace{(2\pi)^{-d/2}}_{h_{\text{N}}(x)} \exp\Big\{\Big\langle \underbrace{(-\tfrac{1}{2}\Sigma^{-1}, \Sigma^{-1}\mu, -\tfrac{1}{2}\mu^\mathsf{T}\Sigma^{-1}\mu)}_{\eta_{\text{N}}(\mu,\Sigma)}, \underbrace{(xx^\mathsf{T}, x, 1)}_{t_{\text{N}}(x)} \Big\rangle - \underbrace{\tfrac{1}{2}\ln|\Sigma|}_{Z_{\text{N}}(\eta_{\text{N}}(\mu,\Sigma))} \Big\} \\ &= \underbrace{(2\pi)^{-d/2}}_{h_{\text{N}}(x)} \exp\Big\{\Big\langle \underbrace{(-\tfrac{1}{2}\Sigma^{-1}, \Sigma^{-1}\mu, -\tfrac{1}{2}\mu^\mathsf{T}\Sigma^{-1}\mu, -\tfrac{1}{2}\ln|\Sigma|)}_{(\eta_{\text{N}}(\mu,\Sigma), -Z_{\text{N}}(\eta_{\text{N}}(\mu,\Sigma)))}, \underbrace{(xx^\mathsf{T}, x, 1, 1)}_{(t_{\text{N}}(x),1)} \Big\rangle\Big\} \end{aligned}$$

*The joint likelihood of n independent samples* $\{x_i \overset{iid}{\sim} \mathcal{N}(\mu, \Sigma) : i \in [n]\}$ *is*

$$\begin{aligned} p(\{x\}_{i=1}^n | \mu, \Sigma) &= \prod_{i=1}^n p(x_i|\mu, \Sigma) \\ &= (2\pi)^{-nd/2} \exp\Big\{ -\frac{1}{2} \Big\langle \left(\eta_{\mathrm{N}}(\mu, \Sigma), -Z_{\mathrm{N}}(\eta_{\mathrm{N}}(\mu, \Sigma))\right), \textstyle\sum_{i=1}^n \left(t_{\mathrm{N}}(x_i), 1\right) \Big\rangle \Big\} \end{aligned}$$

*so we see that* $p(\mu, \Sigma | \{x\}_{i=1}^n, \alpha)$ *is in the NIW family with a new natural parameter*

$$\begin{aligned} \eta_{\mathrm{NIW}}(\Lambda_n, \mu_n, \kappa_n, \nu_n) &= \eta_{\mathrm{NIW}}(\Lambda_0, \mu_0, \kappa_0, \nu_0) + \textstyle\sum_{i=1}^n \left(t_{\mathrm{N}}(x_i), 1\right) \\ &= (\Lambda_0 + \kappa_0 \mu_0 \mu_0^\mathsf{T}, \kappa_0 \mu_0, \kappa_0, \nu_0) + (\textstyle\sum_i x_i x_i^\mathsf{T}, \sum_i x_i, n, n) \end{aligned}$$

*where it can be checked by writing* $\eta_{\mathrm{NIW}}^{-1}$ *that*

$$\begin{aligned} \kappa_n &= \kappa_0 + n \\ \nu_n &= \nu_0 + n \\ \mu_n &= \frac{\kappa_0}{\kappa_0 + n} + \frac{n}{\kappa_0 + n} \bar{x} \\ \Lambda_n &= \Lambda_0 + S + \frac{\kappa_0 n}{\kappa_0 + n} (\bar{x} - \mu_0)(\bar{x} - \mu_0)^\mathsf{T} \end{aligned}$$

*with* $\bar{x} = \frac{1}{n} \sum_i x_i$ *and* $S = \sum_i (x_i - \bar{x})(x_i - \bar{x})^\mathsf{T}$ *as in Gelman et al. [38].*

**Example 2.2.4** (Dirichlet-categorical conjugacy). *Here we show that the Dirichlet is a conjugate prior family for the categorical likelihood of Example 2.2.2.*

*The K-dimensional Dirichlet density with parameter* $\alpha \in \mathbb{R}_+^K$ *where* $\alpha > 0$ *can be written*

$$p(\pi|\alpha) = \mathrm{Dir}(\alpha) = \frac{\Gamma(\sum_{i=1}^K \alpha_i)}{\prod_{i=1}^K \Gamma(\alpha_i)} \prod_{i=1}^K x_i^{\alpha_i - 1} \tag{2.2.27}$$

$$\propto \exp\{\langle \alpha - 1, \ln \pi \rangle\}. \tag{2.2.28}$$

*Using the categorical density we have*

$$p(\pi|x, \alpha) \propto p(\pi|\alpha) p(x|\pi) \tag{2.2.29}$$

$$\propto \exp\{\langle \alpha - 1, \ln \pi \rangle\} \exp\{\langle \ln \pi, \mathbb{1}_x \rangle\} \tag{2.2.30}$$

$$\propto \exp\{\langle \alpha - 1 + \mathbb{1}_x, \ln \pi \rangle\} \tag{2.2.31}$$

*where, as in Example 2.2.2,* $\mathbb{1}_x$ *is an indicator vector with its* $x$*th entry set to* 1 *and its other entries* 0. *Therefore the posterior* $p(\pi|x, \alpha)$ *is in the Dirichlet family with*

**Algorithm 2.1** Gibbs sampling

**Input:** distribution $X$ on graph $G$ with $N$ nodes, conditionals $p_{X_i|X_{\mathrm{MB}_G(i)}}(x_i|x_{\mathrm{MB}_G(i)})$

**Output:** samples $\{\hat{x}^{(t)}\}$

```
Initialize x = (x_1, x_2, ..., x_N)
for t = 1, 2, ... do
    for i = 1, 2, ..., N do
        x_i ~ p_{X_i|X_{MB_G(i)}}( · |x_{MB_G(i)})
    x̂^(t) ← (x_1, x_2, ..., x_N)
```

*parameter* $\alpha + \mathbb{1}_x$. *Similarly, for the Multinomial likelihood*

$$p(x|\pi) = \exp\{\langle \ln \pi, n_x \rangle\} \tag{2.2.32}$$

*where* $n_x = \sum_i \mathbb{1}_{x_i}$ *so that the* $j$*th component is the number of occurences of outcome* $j$, *the posterior* $p(\pi|x, \alpha)$ *is in the Dirichlet family with parameter* $\alpha + n_x$.

## ■ 2.3 Bayesian inference algorithms in graphical models

Here we outline the standard Bayesian inference algorithms and how they relate to the graphical model structure described in Section 2.1 as well as the exponential family conjugacy structure described in Section 2.2. In particular, we describe algorithms that are *compositional* in terms of graphical model structure and that have particularly efficient updates when the graphical model is itself composed of tractable exponential family distributions.

### ■ 2.3.1 Gibbs sampling

In Gibbs sampling, and sampling methods more generally, the task is to generate samples from a distribution of interest so that any probability or statistic can be estimated using the sample population. For a Markov Chain Monte Carlo (MCMC) method such as Gibbs sampling, to generate samples for some collection of random variables $X$ the algorithm simulates a Markov chain on the range of $X$ such that the *limiting distribution* or *stationary distribution* of the chain is the target distribution of $X$. In the Bayesian context, the distribution of interest is typically an intractable posterior.

Given a collection of $n$ random variables $X = \{X_i : i \in [n]\}$, the Gibbs sampling algorithm iteratively samples each variable conditioned on the sampled values of the others. When the random variables are Markov on a graph $G = (V, E)$, the conditioning can be reduced to each variable's respective Markov blanket, as in Algorithm 2.1.

A variant of the *systematic scan* of Algorithm 2.1, in which nodes are traversed in

a fixed order for each outer iteration, is the *random scan*, in which nodes are traversed according to a random permutation sampled for each outer iteration. An advantage of the random scan (and other variants) is that the chain becomes reversible and therefore simpler to analyze [94, Section 10.1.2]. With the conditional independencies implied by a graph, some sampling steps may be performed in parallel.

A Markov chain is called *ergodic* if has a unique steady-state distribution for any initial state; see Robert and Casella [94, Section 6.6.1] for a definition for countable state spaces and Meyn and Tweedie [76, Chapter 13] for a definition for general state spaces. If the Markov chain produced by a Gibbs sampling algorithm is ergodic, then the stationary distribution is the target distribution of $X$ [94, Theorem 10.6]. The Markov chain for a Gibbs sampler can fail to be ergodic if, for example, the support of the target distribution is disconnected [94, Example 10.7]. Many of the Gibbs samplers we develop result in ergodic chains because all of the conditional densities exist and are positive [94, Theorem 10.8]. The main performance criterion of an MCMC sampler is its *mixing time* [94, Chapter 12], which measures the rate at which the distribution of the chain's state reaches the target distribution.

For a more detailed treatment of Gibbs sampling theory, see Robert and Casella [94, Chapter 6, Chapter 10]. For a detailed treatment of Markov chain ergodic theory for general state spaces, as required in precise treatments of Gibbs samplers for the Dirichlet process, see Meyn and Tweedie [76].

### ■ 2.3.2 Mean field variational inference

In mean field, and variational inference more generally, the task is to approximate an intractable distribution, such as a complex posterior, with a distribution from a tractable family in which inference can be performed efficiently. In this section we define the mean field optimization problem and derive the standard coordinate optimization algorithm. We also give some basic results on the relationship between mean field and both graphical model and exponential family structure. For concreteness and simpler notation, we work mostly with undirected graphical models; the results extend immediately to directed models.

Mean field inference makes use of several densities and distributions, and so we use a subscript notation for expectations to clarify the measure used in the integration when it cannot easily be inferred from context. Given a function $f$ and a random variable $X$ with range $\mathcal{X}$ and density $p$ with respect to a base measure $\nu$, we write the expectation of $f$ as

$$\mathbb{E}_{p(X)}\left[f(X)\right] = \int_{\mathcal{X}} f(x)p(x)\nu(dx). \tag{2.3.1}$$

**Proposition 2.3.1** (Variational inequality). *For a density $p$ with respect to a base*

*measure $\nu$ of the form*

$$p(x) = \frac{1}{Z}\bar{p}(x) \quad \textit{with} \quad Z \triangleq \int \bar{p}(x)\nu(dx), \tag{2.3.2}$$

*for all densities $q$ with respect to $\nu$ we have*

$$\ln Z = \mathcal{L}[q] + \mathrm{KL}(q||p) \geq \mathcal{L}[q] \tag{2.3.3}$$

*where*

$$\mathcal{L}[q] \triangleq \mathbb{E}_{q(X)}\left[\ln \frac{\bar{p}(X)}{q(X)}\right] = \mathbb{E}_{q(X)}\left[\ln \bar{p}(X)\right] + \mathbb{H}[q] \tag{2.3.4}$$

$$\mathrm{KL}(q||p) \triangleq \mathbb{E}_{q(X)}\left[\ln \frac{q(X)}{p(X)}\right]. \tag{2.3.5}$$

*Proof.* To show the equality, with $X \sim q$ we write

$$\mathcal{L}[q] + \mathrm{KL}(q||p) = \mathbb{E}_{q(X)}\left[\frac{\bar{p}(X)}{q(X)}\right] + \mathbb{E}_{q(X)}\left[\ln \frac{q(X)}{p(X)}\right] \tag{2.3.6}$$

$$= \mathbb{E}_{q(X)}\left[\ln \frac{\bar{p}(X)}{p(X)}\right] \tag{2.3.7}$$

$$= \ln Z. \tag{2.3.8}$$

The inequality follows from the property $\mathrm{KL}(q||p) \geq 0$, known as Gibbs's inequality, which follows from Jensen's inequality and the fact that the logarithm is concave:

$$-\mathrm{KL}(q||p) = \mathbb{E}_{q(X)}\left[\ln \frac{q(X)}{\bar{p}(X)}\right] \leq \ln \int q(x)\frac{\bar{p}(x)}{q(x)}\nu(dx) = 0 \tag{2.3.9}$$

with equality if and only if $q = p$ ($\nu$-a.e.). □

We call the log of $\bar{p}$ in (2.3.2) the *energy* and $\mathcal{L}[q]$ the *variational lower bound*, and say $\mathcal{L}[q]$ decomposes into the average energy plus entropy as in (2.3.4). For two densities $q$ and $p$ with respect to the same base measure, $\mathrm{KL}(q||p)$ is the *Kullback-Leibler divergence* from $q$ to $p$, used as a measure of dissimilarity between pairs of densities [2].

The variational inequality given in Proposition 2.3.1 is useful in inference because if we wish to approximate an intractable $p$ with a tractable $q$ by minimizing $\mathrm{KL}(q||p)$, we can equivalently choose $q$ to maximize $\mathcal{L}[q]$, which is possible to evaluate since it does not include the partition function $Z$.

In the context of Bayesian inference, $p$ is usually an intractable posterior distribution

of the form $p(\theta|x,\alpha)$, $\bar{p}$ is the unnormalized product of the prior and likelihood $\bar{p}(\theta) = p(\theta|\alpha)p(x|\theta)$, and $Z$ is the marginal likelihood $p(x|\alpha) = \int p(x|\theta)p(\theta|\alpha)\nu(d\theta)$, which plays a central role in Bayesian model selection and the minimum description length (MDL) criterion [74, Chapter 28] [49, Chapter 7].

Given that graphical model structure can affect the complexity of probabilistic inference, as discussed in Section 2.1.3, it is natural to consider families $q$ that factor according to tractable graphs.

**Definition 2.3.1** (Mean field variational inference). *Let $p$ be the density with respect to $\nu$ for a collection of random variables $X = (X_i : i \in V)$, and let*

$$\mathcal{Q} \triangleq \{q : q(x) \propto \prod_{C \in \mathcal{C}} q_C(x_C)\} \tag{2.3.10}$$

*be a family of densities with respect to $\nu$ that factorize according to a graph $G = (V, E)$ with $\mathcal{C}$ the set of maximal cliques of $G$. Then the* mean field optimization problem *is*

$$q^* = \arg\max_{q \in \mathcal{Q}} \mathcal{L}[q] \tag{2.3.11}$$

*where $\mathcal{L}[q]$ is defined as in* (2.3.4).

Note that the optimization problem is not convex and so one can only expect to find a local optimum of the objective [113]. However, since the objective is convex in each $q_C$ individually, an optimization procedure that updates each factor in turn holding the rest constant will converge to a local optimum [11, Chapter 10]. We call such a coordinate ascent procedure on (2.3.11) a *mean field algorithm.* For approximating families in a factored form, we can derive a generic update to be used in a mean field algorithm.

**Proposition 2.3.2** (Mean field update). *Given a mean field objective as in Definition 2.3.1, the optimal update to a factor $q_A$ fixing the other factors defined by $q_A^* = \arg\max_{q_A} \mathcal{L}[q]$ is*

$$q_A^*(x_A) \propto \exp\{\mathbb{E}[\ln \bar{p}(x_A, X_{A^c})]\} \tag{2.3.12}$$

*where the expectation is over $X_{A^c} \sim q_{A^c}$ with $q_{A^c}(x_{A^c}) \propto \prod_{C \in \mathcal{C} \setminus A} q_C(x_C)$.*

*Proof.* Dropping terms constant with respect to $q_A$, we write

$$q_A^* = \arg\min_{q_A} \mathrm{KL}(q||p) \tag{2.3.13}$$

$$= \arg\min_{q_A} \mathbb{E}_{q_A}[\ln q_A(X_A)] + \mathbb{E}_{q_A}[\mathbb{E}_{q_{A^c}}[\log \bar{p}(X)]] \tag{2.3.14}$$

$$= \arg\min_{q_A} \mathrm{KL}(q_A||\widetilde{p}_A) \tag{2.3.15}$$

where $\widetilde{p}_A(x_A) \propto \exp\{\mathbb{E}_{q_{A^c}}[\ln \bar{p}(x_A, X_{A^c})]\}$. Therefore, we achieve the unique ($\nu$-a.e.) minimum by setting $q_A = \widetilde{p}_A$. □

Furthermore, if $p$ factors according to a graph then the same graph structure is induced in the factors of the optimal $q$.

**Proposition 2.3.3** (Induced graph structure). *If $p$ is Markov on a graph $G = (V, E)$ with the form $p(x) \propto \prod_{C \in \mathcal{C}} \psi_C(x_C)$ for cliques $\mathcal{C}$ of $G$, then any optimal factor $q_A$ with $A \subseteq V$ factors according to $G \setminus A^c$. Furthermore, the update (2.3.12) can be computed using only the factors on the cliques $\mathcal{C}' = \{C \in \mathcal{C} : C \cap A \neq \varnothing\}$, i.e. the cliques on variables in the Markov blanket of $A$.*

*Proof.* Using (2.3.12) we have

$$q_A^*(x_A) \propto \exp\{\mathbb{E}[\ln \bar{p}(x_A, X_{A^c})]\} \propto \exp\left\{\sum_{C \in \mathcal{C}'} \mathbb{E}[\ln \psi_C(X_C)]\right\} \tag{2.3.16}$$

where factors not involving the variables in $A$ are dropped up to proportionality. □

Because $q$ inherits the graphical structure of $p$, it is therefore natural to consider tractable families $\mathcal{Q}$ that are Markov on *subgraphs* of $p$. Note that when $q$ is a subgraph of $p$, the variational lower bound is a sum of terms corresponding to the factors of $p$. When the family $\mathcal{Q}$ is chosen to be Markov on the completely disconnected graph $G = (V, E)$ with $E = \varnothing$, the resulting algorithm is called *naive mean field*. When the tractable subgraph retains some nontrivial graphical structure, the algorithm is called *structured mean field*. In this thesis we use structured mean field extensively for inference in time series models.

Finally, we note the simple form of updates for exponential family conjugate pairs.

**Proposition 2.3.4** (Mean field and conjugacy). *If $x_i$ appears in $\bar{p}$ only in an exponential family conjugate pair $(p_1, p_2)$ where*

$$p_1(x_i | x_{\pi_G(i)}) \propto \exp\{\langle \eta(x_{\pi_G(i)}), t(x_i) \rangle\} \tag{2.3.17}$$

$$p_2(x_{c_G(i)} | x_i) = \exp\{\langle t(x_i), (t(x_{c_G(i)}), 1) \rangle\} \tag{2.3.18}$$

*then the optimal factor $q_i(x_i)$ is in the prior family with natural parameter*

$$\widetilde{\eta} \triangleq \mathbb{E}_q[\eta(X_{\pi_G(i)})] + \mathbb{E}_q[(t(X_{c_G(i)}), 1)]. \tag{2.3.19}$$

*Proof.* The result follows from substituting (2.3.17) and (2.3.18) into (2.3.12). □

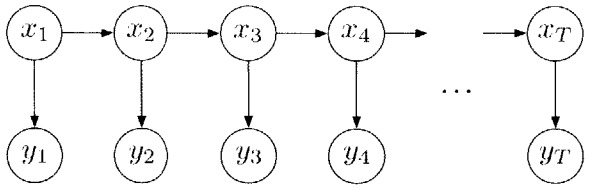



**Figure 2.3:** Directed graphical model for an HMM.

See Wainwright and Jordan [113, Chapter 5] for a convex analysis perspective on mean field algorithms in graphical models composed of exponential families.

## ■ 2.4 Hidden Markov Models

In this section we use the definitions and results from previous sections to define the Bayesian Hidden Markov Model and give Gibbs sampling and mean field inference algorithms. For simplicity, we refer only to a single observation sequence; the generalization to multiple observation sequences is immediate.

A Hidden Markov Model (HMM) on $N$ states defines a joint distribution over a *state sequence* $x_{1:T}$ and an *observation sequence* $y_{1:T}$. It is parameterized by an *initial state distribution* $\pi^{(0)} \in \mathbb{R}_+^N$, a *transition matrix* $A \in \mathbb{R}_+^{N \times N}$, and emission parameters $\theta = \{\theta^{(i)}\}_{i=1}^N$. We use $\pi^{(i)}$ to denote the $i$th row of $A$ and collect the transition rows and initial state distribution into $\pi = \{\pi^{(i)}\}_{i=0}^N$ for convenient notation.

Recall that for two random variables $X$ and $Y$, we write $X \sim Y$ to denote that $X$ and $Y$ have the same distribution. For three random variables $X$, $Y$, and $Z$, we similarly write $X|Y \sim Z$ to denote that $X|Y$ has the same distribution as $Z$. Finally, recall that for a random variable $X$ and a density $p$ we write $X \sim p(\,\cdot\,)$ to denote that $X$ has density $p$. We use this notation to specify generative models, as in the following definition.

**Definition 2.4.1.** *We say sequences of random variables $(x_{1:T}, y_{1:T})$ are distributed according to a Hidden Markov Model, and write $(x_{1:T}, y_{1:T}) \sim HMM(\pi, \theta)$, when they follow the generative process*

$$x_1 \sim \pi^{(0)}, \tag{2.4.1}$$

$$x_{t+1}|x_t \sim \pi^{(x_t)} \qquad t = 1, 2, \ldots, T-1, \tag{2.4.2}$$

$$y_t|x_t \sim p(\,\cdot\,|\theta^{(x_t)}) \qquad t = 1, 2, \ldots, T. \tag{2.4.3}$$

*Figure 2.3 shows a graphical model for the HMM.*

If each $p(y|\theta^{(i)})$ is an exponential family of densities in natural parameters of the

**Algorithm 2.2** HMM Forwards Messages

**Input:** transition potentials $A$, emission potentials $L$, initial state potential $\pi^{(0)}$
**Output:** HMM forwards messages $F$
**function** HMMFWDMESSAGES($A$,$L$,$\pi^{(0)}$)
  $F_{1,:} \leftarrow \pi^{(0)} \odot L_{1,:}$
  **for** $t = 2, 3, \ldots, T$ **do**
    $F_{ti} \leftarrow \sum_{j=1}^{N} F_{t-1,j} A_{ji} L_{ti}$
  **return** $F$

**Algorithm 2.3** HMM Backwards Messages

**Input:** transition potentials $A$, emission potentials $L$
**Output:** HMM backwards messages $B$
**function** HMMBWDMESSAGES($A$,$L$)
  $B_{T,:} \leftarrow 1$
  **for** $t = T-1, T-2, \ldots, 1$ **do**
    $B_{t,i} \leftarrow \sum_{j=1}^{N} A_{ij} B_{t+1,j} L_{t+1,j}$
  **return** $B$

form

$$p(y|\eta^{(i)}) = \exp\{\langle \eta_y^{(i)}, t_y^{(i)}(y) \rangle - Z_y(\eta_y^{(i)})\} \tag{2.4.4}$$

then we can write the joint density as an exponential family:

$$p(x_{1:T}, y_{1:T}) = \exp\left\{ (\ln \pi^{(0)})^\mathsf{T} \mathbb{1}_{x_1} + \sum_{t=1}^{T-1} \mathbb{1}_{x_t}^\mathsf{T} A \mathbb{1}_{x_{t+1}} + \sum_{t=1}^{T} \langle \eta_y^{(i)}, \mathbb{I}[x_t = i] \cdot t_y^{(i)}(y_t) \rangle \right\}. \tag{2.4.5}$$

Since the HMM is Markov on an undirected tree (more precisely, a chain), we can use the tree message-passing algorithm to perform inference efficiently. The HMM messages and recursions are typically written in terms of *forward messages* $F$ and *backward messages* $B$, where

$$F_{t,i} \triangleq p(y_{1:t}, x_t) = \sum_{j=1}^{N} A_{ji} F_{t-1,j} p(y_t|\theta^{(i)}) \tag{2.4.6}$$

$$B_{t,i} \triangleq p(y_{t+1:T}|x_t = i) = \sum_{j=1}^{N} A_{ij} p(y_{t+1}|\theta^{(j)}) B_{t+1,j} \tag{2.4.7}$$

with the initial values $F_{1,i} = \pi_i^{(0)} p(y_1|\theta^{(i)})$ and $B_{T,i} = 1$. Algorithms to compute these messages are given in Algorithms 2.2 and 2.3.

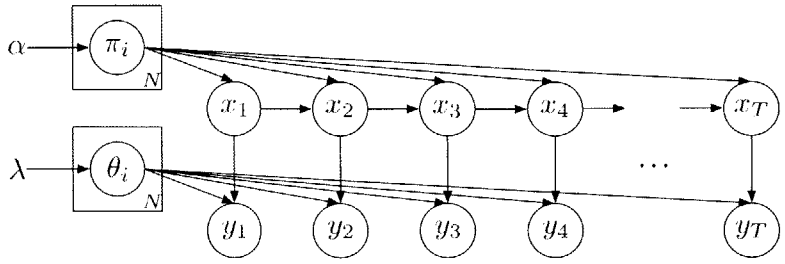



**Figure 2.4:** Directed graphical model for a Bayesian HMM.

A Bayesian treatment of HMMs places priors on the parameters $(\pi, \theta)$ and includes them in the probabilistic model. We use $\text{Dir}(\alpha)$ where $\alpha \in \mathbb{R}_+^K$ for some $K > 0$ to denote the Dirichlet distribution with parameter $\alpha$.

**Definition 2.4.2.** *We say $(\theta, \pi, x_{1:T}, y_{1:T})$ are distributed according to a Bayesian Hidden Markov Model with hyperparameters $\alpha = \{\alpha^{(i)}\}_{i=1}^N$ and $\lambda = \{\lambda^{(i)}\}_{i=1}^N$, and write $(\theta, \pi, x_{1:T}, y_{1:T}) \sim BayesHMM(\alpha, \lambda)$, when they follow the generative process*

$$\pi^{(i)} \overset{iid}{\sim} Dir(\alpha^{(i)}) \tag{2.4.8}$$

$$\theta^{(i)} \overset{iid}{\sim} p(\,\cdot\,|\lambda^{(i)}) \tag{2.4.9}$$

$$(x_{1:T}, y_{1:T})|\pi, \theta \sim HMM(\pi, \theta) \tag{2.4.10}$$

*where $HMM(\pi, \theta)$ is defined in Definition 2.4.1. Figure 2.4 shows a graphical model for the Bayesian HMM.*

Given an observation sequence $\bar{y}_{1:T}$ the task of interest is to perform inference in the posterior $\pi, \theta, x_{1:T}|\bar{y}_{1:T}$. Note In this section we develop both Gibbs sampling and mean field algorithms for this inference task.

## ■ 2.4.1 HMM Gibbs sampling

The HMM Gibbs sampler iterates sampling $\theta$, $\pi$, and $x_{1:T}$ from their respective conditionals. To sample the state sequence $x_{1:T}$ from its conditional, we exploit the tree message-passing algorithm. Furthermore, since the Dirichlet is the conjugate prior to the categorical from which each state is sampled, the conditional for $\pi$ is also Dirichlet. The HMM Gibbs sampler can then be written as Algorithm 2.4.

An alternative Gibbs sampler can be constructed by marginalizing the parameters $(\pi, \theta)$, which is tractable when the observation prior and likelihood form a conjugate pair, and generating samples of $x_{1:T}|\bar{y}_{1:T}, \alpha, \lambda$. While such collapsed samplers can be advantageous in some settings, in the case of a Bayesian HMM eliminating $\pi$ and $\theta$

**Algorithm 2.4** Bayesian HMM Gibbs sampling

**Input:** $\alpha$, $\lambda$, $\bar{y}_{1:T}$
**Output:** samples $\{(\hat{x}_{1:T}, \hat{\theta}, \hat{\pi})^{(t)}\}$
  Initialize $x_{1:T}$
  **for** $t = 1, 2, \ldots$ **do**
    **for** $i = 1, 2, \ldots, N$ **do**
      $\pi^{(i)} \leftarrow$ sample $\mathrm{Dir}(\alpha^{(i)} + n_{i:})$ with $n_{ij} = \sum_t \mathbb{I}[x_t = i, x_{t+1} = j]$
      $\theta^{(i)} \leftarrow$ sample $p(\theta|\lambda, \{y_t : x_t = i\})$
    $\pi^{(0)} \leftarrow$ sample $\mathrm{Dir}(\alpha^{(0)} + \mathbb{1}_{x_1})$
    $x_{1:T} \leftarrow$ HMMSAMPLESTATES$(\pi^{(0)}, A, L)$ with $L_{t,i} = p(y_t|\theta^{(i)})$
    $(\hat{x}_{1:T}, \hat{\theta}, \hat{\pi})^{(t)} \leftarrow (x_{1:T}, \theta, \pi)$

**Algorithm 2.5** HMM state sequence sampling

**Input:** $\pi^{(0)}$, $A$, $L$
**Output:** a sample $x_{1:T}$
  **function** HMMSAMPLESTATES($A$,$L$)
    $B \leftarrow$ HMMBWDMESSAGES$(A, L)$
    $x_1 \leftarrow$ sample $\pi_i^{(0)} B_{1,i} L_{1,i}$ over $i$
    **for** $t = 2, 3, \ldots, T$ **do**
      $x_t \leftarrow$ sample $A_{x_{t-1},i} B_{t,i} L_{t,i}$ over $i$

induces a full graph on the remaining nodes, and so one cannot exploit tree message passing to construct a joint sample of the state sequence and each $x_t$ must be resampled one at a time. Because the $x_t$ are highly correlated in the model, the collapsed Gibbs sampler is often slow to explore the posterior.

### ■ 2.4.2 HMM Mean Field

Here we briefly overview a mean field algorithm for HMMs. For more details on the HMM mean field algorithm, see Beal [6, Chapter 3].

We choose a variational family that factorizes as $q(\pi, \theta, x_{1:T}) = q(\pi, \theta)q(x_{1:T})$ so that the parameters and state sequence are decoupled. The Bayesian HMM graphical model then induces independences so that the variational family is

$$q(\pi, \theta, x_{1:T}) = \prod_{i=0}^{N} q(\pi^{(i)}) q(\theta^{(i)}) q(x_{1:T}). \tag{2.4.11}$$

Note that because $\pi^{(i)}$ is the $i$th row of the transition matrix $A$, $i = 1, 2, \ldots, N$, we write $q(\pi^{(1)}, \ldots, \pi^{(N)})$ equivalently as $q(A)$ to simplify some notation. The variational objective function is

$$\mathbb{E}\left[\ln \frac{p(\pi,\theta,x,y)}{q(\pi)q(\theta)q(x_{1:T})}\right] = \mathbb{E}_{q(\pi)}\left[\ln \frac{p(\pi)}{q(\pi)}\right] + \mathbb{E}_{q(\theta)}\left[\ln \frac{p(\theta)}{q(\pi)}\right] \tag{2.4.12}$$

$$+ \mathbb{E}_{q(x_{1:T})q(\pi)q(\theta)}\left[\ln \frac{p(x_{1:T}, y_{1:T}|\pi,\theta)}{q(x_{1:T})}\right]. \tag{2.4.13}$$

For more explicit updates, we assume that each observation prior and likelihood form a conjugate pair of densities and that the prior family is written in natural parameters with the form

$$p(\theta^{(i)}|\eta_\theta^{(i)}) \propto \exp\{\langle \eta_\theta^{(i)}, t_\theta^{(i)}(\theta^{(i)})\rangle\} \qquad p(y|\theta^{(i)}) = \exp\{\langle t_\theta^{(i)}(\theta^{(i)}), (t_y(y), 1)\rangle\}. \tag{2.4.14}$$

Then the update for the factor $q(x_{1:T})$ is

$$q^*(x_{1:T}) \propto \mathbb{E}[\ln p(x_{1:T}, y_{1:T}|\theta,\pi)] \tag{2.4.15}$$

$$= \exp\left\{ (\mathbb{E}_{q(\pi)}[\ln \pi^{(0)}])^\top \mathbb{1}_{x_1} + \sum_{t=1}^{T} \mathbb{1}_{x_t}^\top \mathbb{E}_{q(A)}[\ln A]\mathbb{1}_{x_{t+1}} \right.$$

$$\left. + \sum_{t=1}^{T} \langle \mathbb{E}_{q(\theta)}[t_\theta^{(i)}(\theta^{(i)})],\ \mathbb{I}[x_t = i] t_y(y_t, 1)\rangle \right\} \tag{2.4.16}$$

and so, as expected, the optimal factor is also Markov on a chain graph.

For the conjugate updates to $q(\pi)$ and $q(\theta)$, we write the variational factors as

$$q(\theta^{(i)}) \propto \exp\{\langle \widetilde{\eta}_\theta^{(i)}, t_\theta^{(i)}(\theta^{(i)})\rangle\} \qquad q(\pi^{(i)}) = \mathrm{Dir}(\widetilde{\alpha}^{(i)}). \tag{2.4.17}$$

We can compute the expected sufficient statistics over $q(x_{1:T})$ by running the HMM message-passing algorithm. Defining

$$\widetilde{\pi}^{(i)} \triangleq \mathbb{E}_{q(\pi)}[\ln \pi^{(i)}] \qquad \widetilde{L}_{t,i} \triangleq \mathbb{E}_{q(\theta)}[\ln p(y_t|\theta^{(i)})], \tag{2.4.18}$$

and defining $\widetilde{A}$ to be a matrix where the $i$th row is $\widetilde{\pi}^{(i)}$ for $i = 1, 2, \ldots, N$, we compute

**Algorithm 2.6** HMM Mean Field

Initialize variational parameters $\widetilde{\eta}_\theta^{(i)}$, $\widetilde{\alpha}^{(i)}$, $\widetilde{\alpha}^{(0)}$
**for** $t = 1, 2, \ldots$ until convergence **do**
  $F \leftarrow$ HMMFwdMessages$(\widetilde{A}, \widetilde{L}, \widetilde{\pi}^{(0)})$
  $B \leftarrow$ HMMBwdMessages$(\widetilde{A}, \widetilde{L})$
  Using $F$ and $B$, compute each $\hat{t}_y^{(i)}$, $\hat{t}_{\text{trans}}^{(i)}$, and $\hat{t}_{\text{init}}$
    with Eqs. (2.4.19)-(2.4.21)
  Update $\widetilde{\eta}_\theta^{(i)}$, $\widetilde{\alpha}^{(i)}$, $\widetilde{\alpha}^{(0)}$ for $i = 1, 2, \ldots, N$
    with Eqs. (2.4.22)-(2.4.24)

$$\hat{t}_y^{(i)} \triangleq \mathbb{E}_{q(x_{1:T})} \sum_{t=1}^{T} \mathbb{1}[x_t = i] t_y^{(i)}(\bar{y}_t) = \sum_{t=1}^{T} F_{t,i} B_{t,i} \cdot (t_y^{(i)}(\bar{y}_t), 1)/Z \tag{2.4.19}$$

$$(\hat{t}_{\text{trans}}^{(i)})_j \triangleq \mathbb{E}_{q(x_{1:T})} \sum_{t=1}^{T-1} \mathbb{1}[x_t = i, x_{t+1} = j] = \sum_{t=1}^{T-1} F_{t,i} \widetilde{A}_{ij} \widetilde{L}_{t+1,j} B_{t+1,j}/Z \tag{2.4.20}$$

$$(\hat{t}_{\text{init}})_i \triangleq \mathbb{E}_{q(x_{1:T})} \mathbb{1}[x_1 = i] = \widetilde{\pi}_0 B_{1,i}/Z \tag{2.4.21}$$

where $Z = \sum_{i=1}^{N} F_{T,i}$. With these expected statistics, the updates to the parameters of $q(A)$, $q(\pi_0)$, and $q(\theta)$ are then

$$\widetilde{\eta}_\theta^{(i)} \leftarrow \eta_\theta^{(i)} + \hat{t}_y^{(i)} \tag{2.4.22}$$

$$\widetilde{\alpha}^{(i)} \leftarrow \alpha^{(i)} + \hat{t}_{\text{trans}}^{(i)} \tag{2.4.23}$$

$$\widetilde{\alpha}^{(0)} \leftarrow \alpha^{(0)} + \hat{t}_{\text{init}}^{(i)}. \tag{2.4.24}$$

We summarize the overall algorithm in Algorithm 2.6.

## ■ 2.5 The Dirichlet Process and Nonparametric Models

The Dirichlet process is used to construct Bayesian nonparametric models, including nonparametric HMMs such that the number of states is unbounded a priori. Bayesian nonparametric methods allow model complexity to be learned flexibly from data and to grow as the amount of data increases. In this section, we review the basic definition of the Dirichlet process and the HDP-HMM.

**Definition 2.5.1** (Dirichlet process). *Let $(\Omega, \mathcal{F}, H)$ be a probability space and $\alpha > 0$. We say $G$ is distributed according to a Dirichlet process with parameter $\alpha H$, and write $G \sim \text{DP}(\alpha H)$ or $G \sim \text{DP}(\alpha, H)$, if $(\Omega, \mathcal{F}, G)$ is a probability space and for every finite*

*partition* $\{A_i \subseteq \Omega : i \in [r]\}$ *of* $\Omega$

$$\bigcup_{i=1}^{r} A_i = \Omega \qquad i \neq j \implies A_i \cap A_j = \varnothing, \tag{2.5.1}$$

*we have*

$$(G(A_1), \ldots, G(A_r)) \sim \mathrm{Dir}(\alpha H(A_1), \ldots, \alpha H(A_r)). \tag{2.5.2}$$

By the Kolmogorov consistency theorem, this definition in terms of consistent finite-dimensional marginals defines a unique stochastic process [88]. Though the definition is not constructive, some properties of the Dirichlet process are immediate.

**Proposition 2.5.1.** *If* $G \sim \mathrm{DP}(\alpha H)$, *then*

1. $G$ *is* atomic *w.p. 1, meaning it can be written*

$$G = \sum_{i=1}^{\infty} \pi_i \delta_{\theta_i} \tag{2.5.3}$$

*for some atoms* $\omega_i \in \Omega$ *and weights* $\pi_i \in (0, 1)$.

2. *If* $\theta_i | G \overset{iid}{\sim} G$ *for* $i = 1, 2, \ldots, N$, *then* $G | \{\theta_i\}_{i=1}^N$ *is distributed as a Dirichlet process with*

$$G | \{\theta_i\}_{i=1}^N \sim \mathrm{DP}(\alpha H + \sum_{i=1}^{N} \delta_{\theta_i}). \tag{2.5.4}$$

*Proof.* As shown in Ferguson [27], these properties follow from Definition 2.5.1 and finite Dirichlet conjugacy. □

A construction that satisfies Definition 2.5.1 is the *stick breaking process*. In the following, we use $X \sim \mathrm{Beta}(\alpha, \beta)$ to denote that $X$ has the density

$$p(x | \alpha, \beta) \propto x^{\alpha - 1}(1 - x)^{\beta - 1}. \tag{2.5.5}$$

**Definition 2.5.2** (Stick-breaking process). *We say* $\pi = \{\pi_i : i \in \mathbb{N}\}$ *is distributed according to the stick-breaking process with parameter* $\alpha > 0$, *and write* $\pi \sim \mathrm{GEM}(\alpha)$, *if*

$$\beta_i \sim \mathrm{Beta}(1, \alpha), \qquad \pi_i = \beta_i \prod_{j<i} (1 - \beta_j), \qquad i = 1, 2, \ldots. \tag{2.5.6}$$

**Theorem 2.5.1** (Stick-breaking construction). *Let* $\pi \sim \mathrm{GEM}(\alpha)$ *and* $\theta_i \overset{iid}{\sim} H$ *for* $i \in \mathbb{N}$. *If* $G = \sum_{i=1}^{\infty} \pi_i \delta_{\theta_i}$ *then* $G \sim \mathrm{DP}(\alpha H)$.

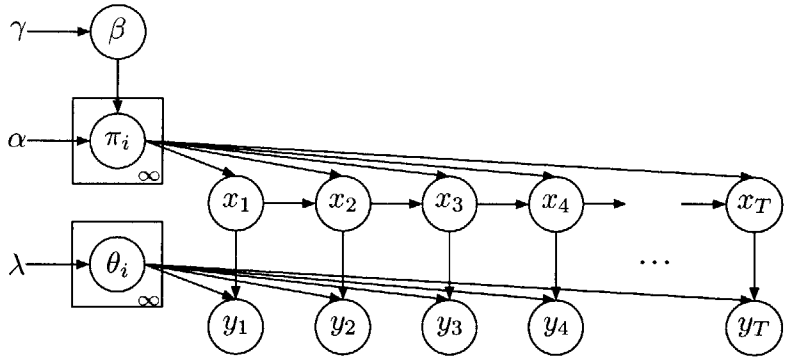



**Figure 2.5:** Directed graphical model for an HDP-HMM.

*Proof.* See Sethuraman [103]. □

We can define Dirichlet processes that share the same set of atoms and have similar weights using the hierarchical Dirichlet process construction [107]. This construction is useful in defining a Bayesian nonparametric extension of the HMM.

**Definition 2.5.3** (Hierarchical Dirichlet process). *We say a collection of random measures $\{G_j : j \in \mathbb{N}\}$ are distributed according to the hierarchical Dirichlet process with parameters $\alpha$, $\gamma$, and $H$ if*

$$G_0 \sim \mathrm{DP}(\alpha, H) \qquad G_j \overset{iid}{\sim} \mathrm{DP}(\gamma, G_0). \tag{2.5.7}$$

We can use the stick-breaking construction of the DP to define a stick-breaking construction of the HDP.

**Definition 2.5.4.** *We say $(\pi, \theta, x_{1:T}, y_{1:T})$ are distributed according to an HDP-HMM with parameters $\alpha, \gamma > 0$ and base measure $H$ if*

$$\beta \sim \mathrm{GEM}(\gamma), \qquad \pi_i \overset{iid}{\sim} \mathrm{DP}(\alpha\beta), \qquad \theta_i \overset{iid}{\sim} H, \tag{2.5.8}$$

$$x_t \sim \pi_{x_{t-1}}, \qquad y_t \sim p(\,\cdot\,|\theta_{x_t}) \tag{2.5.9}$$

*where $\beta$ is treated as a density with respect to counting measure on $\mathbb{N}$ and where we set $x_1 = 0$. We write $(\pi, \theta, x_{1:T}, y_{1:T}) \sim \mathrm{HDP\text{-}HMM}(\alpha, \gamma, H)$.*

*Figure 2.5 shows a graphical model for the HDP-HMM.*

There are several methods to perform sampling inference in Dirichlet process models. First, exploiting the conjugacy properties of the Dirichlet process, one can analytically marginalize the DP draws, as in the Chinese Restaurant Process (CRP) and Chinese Restaurant Franchise (CRF) samplers for the Dirichlet process and hierarchical Dirichlet process, respectively [107]. However, as before, eliminating variables

introduces many dependencies and can result in poor sampler performance for models like the HDP-HMM. One can also work with a finite instantiation of the Dirichlet process draws, so that the sampler only needs to work with the finite Dirichlet marginals, as in the Direct Assignment sampler [107], but such a construction still precludes tree message-passing in an HDP-HMM. The approach we take for most samplers in this thesis is based on approximating a DP prior with a finite symmetric Dirichlet distribution, where the notion of approximation is made precise in the following result.

**Theorem 2.5.2** (Weak limit approximation). *Let $(\Omega, \mathcal{F}, H)$ be a probability space, $\alpha > 0$ be a positive constant, and $f : \Omega \to \mathbb{R}$ be any $(\mathcal{F}, \mathcal{B}(\mathbb{R}))$-measurable function. Consider the finite model of size $K$ given by*

$$\theta_i \overset{iid}{\sim} H \qquad \pi = (\pi_1, \ldots, \pi_K) \sim \mathrm{Dir}(\alpha) \tag{2.5.10}$$

*and define the measure $G^K = \sum_{i=1}^{K} \pi_i \delta_{\theta_i}$. Then as $K \to \infty$ we have*

$$\int f(\omega) G^K(d\omega) \xrightarrow{\mathcal{D}} \int f(\omega) G(d\omega) \tag{2.5.11}$$

*where $G \sim \mathrm{DP}(\alpha H)$.*

*Proof.* See Ishwaran and Zarepour [57, Theorem 2], which also gives rates of convergence and bounds on the probabilities of some error events. □

Based on this approximation result, we can define Bayesian nonparametric models and perform approximate inference with finite models of size $K$, where $K$ becomes an algorithm parameter rather than a model parameter. With these finite approximations we can exploit graphical model structure and tree message-passing algorithms in both Gibbs sampling and mean field algorithms for time series models defined with the HDP.

Chapter 3

# Hierarchical Dirichlet Process Hidden Semi-Markov Models

## 3.1 Introduction

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

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

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

One approach to avoiding the rapid-switching problem is the Sticky HDP-HMM [31],

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

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

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

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

## ■ 3.2 Background and Notation

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

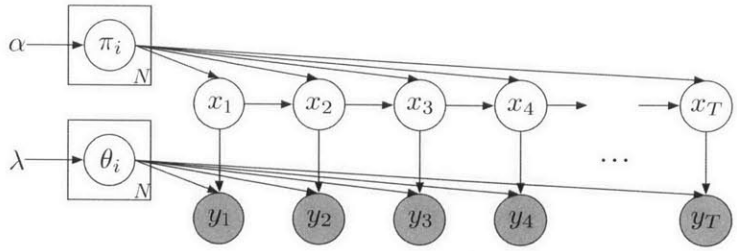



**Figure 3.1:** Basic graphical model for the Bayesian HMM. Parameters for the transition, emission, and initial state distributions are random variables. The symbol $\alpha$ represents the hyperparameter for the prior distributions on state-transition parameters. The shaded nodes indicate observations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

### ■ 3.2.1 HMMs

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

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

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

### ■ 3.2.2 HSMMs

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

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

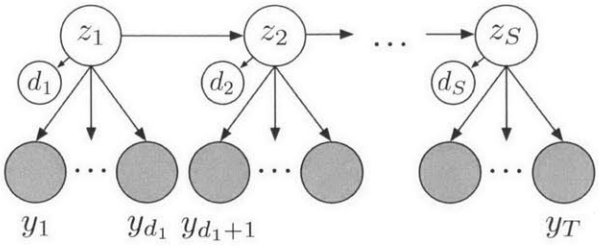



**Figure 3.2:** HSMM interpreted as a Markov chain on a set of super-states, $(z_s)_{s=1}^S$. The number of shaded nodes associated with each $z_s$, denoted by $d_s$, is drawn from a state-specific duration distribution.

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

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

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

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

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

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

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

$$\begin{aligned} B_t(i) &\triangleq p(y_{t+1:T}|x_t = i, x_t \neq x_{t+1}) \\ &= \sum_j B_t^*(j) p(x_{t+1} = j | x_t = i), \end{aligned} \tag{3.2.4}$$

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

$$B_T(i) \triangleq 1, \tag{3.2.6}$$

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

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

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

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

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

### 3.2.3 The HDP-HMM and Sticky HDP-HMM

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

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

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

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

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

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

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

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

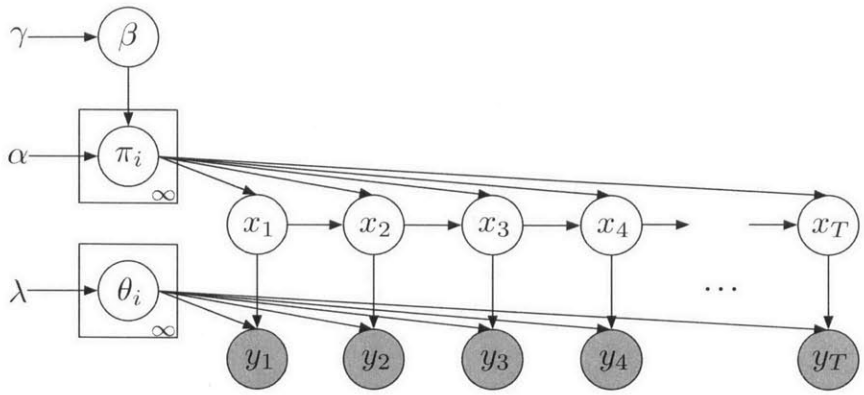



**Figure 3.3:** Graphical model for the HDP-HMM.

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

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

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

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

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

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

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

## 3.3 HSMM Models

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

### 3.3.1 Finite Bayesian HSMM

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

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

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

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

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

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

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

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

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

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

### ■ 3.3.2 HDP-HSMM

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

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

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

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

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

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

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

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

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

### ■ 3.3.3 Factorial Structure

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

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

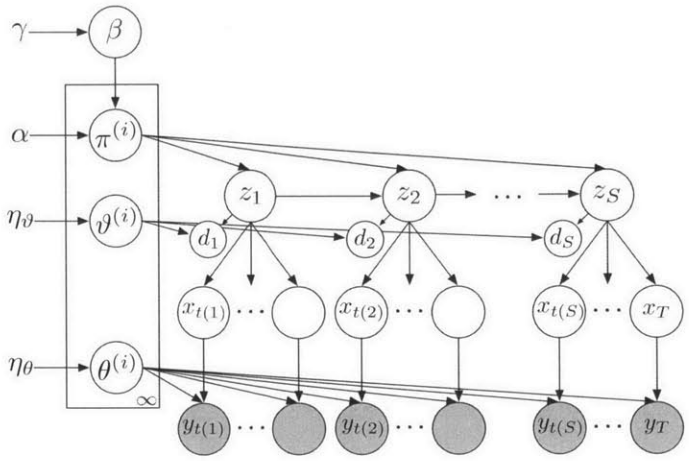



**Figure 3.4:** A graphical model for the HDP-HSMM in which the number of segments $S$, and hence the number of nodes, is random.

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

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

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

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

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

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

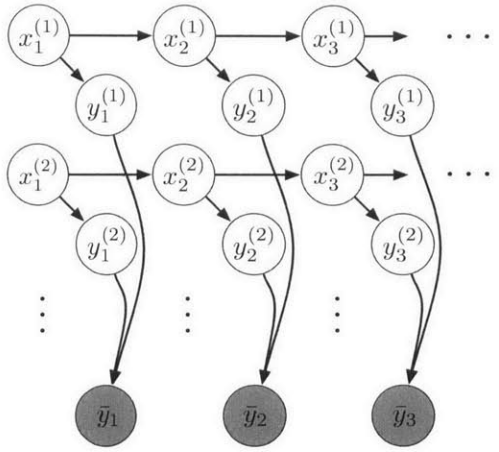



**Figure 3.5:** A graphical model for the factorial HMM, which can naturally be extended to factorial structures involving the HSMM or HDP-HSMM.

## ■ 3.4 Inference Algorithms

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

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

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

### ■ 3.4.1 A Gibbs Sampler for the Finite Bayesian HSMM

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

#### Outline of Gibbs Sampler

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

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

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

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

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

#### Blocked Conditional Sampling of $(x_t)$ with Message Passing

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

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

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

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

We can also use the messages to efficiently draw a sample from the posterior duration

distribution for the sampled initial state. Conditioning on the initial state draw, $\bar{x}_1$, the posterior duration of the first state is:

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

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

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

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

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

### ■ 3.4.2 A Weak-Limit Gibbs Sampler for the HDP-HSMM

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

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

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

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

We can employ the weak limit approximation to create a finite HSMM that approximates inference in the HDP-HSMM. This approximation technique often results in greatly accelerated mixing, and hence it is the technique we employ for the experiments

in the sequel. However, the inference algorithm of Section 3.4.1 must be modified to incorporate the fact that the $\{\pi^{(i)}\}$ are no longer mutually independent and are instead tied through the shared $\beta$. This dependence between the transition rows introduces potential conjugacy issues with the hierarchical Dirichlet prior; the following section explains the difficulty as well as a simple algorithmic solution via auxiliary variables.

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

#### Conditional Sampling of $\{\pi^{(i)}\}$ with Data Augmentation

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

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

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

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

$$\tag{3.4.12}$$

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

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

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

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

$$p(\pi^{(1)}|(x_t),\beta) \propto p(\pi^{(1)}|\beta)p((x_t)|\pi^{(1)}) \tag{3.4.14}$$

$$\propto (\pi_1^{(1)})^{\alpha\beta_1-1}\cdots(\pi_L^{(1)})^{\alpha\beta_L-1}\left(\frac{\pi_2^{(1)}}{1-\pi_1^{(1)}}\right)^{n_{12}}\cdots\left(\frac{\pi_L^{(1)}}{1-\pi_1^{(1)}}\right)^{n_{1L}}, \tag{3.4.15}$$

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

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

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

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

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

$$\pi^{(1)}|\beta \sim \mathrm{Dir}(\beta), \tag{3.4.17}$$

$$z_i|\bar{\pi}^{(1)} \sim \bar{\pi}_1 \tag{3.4.18}$$

$$y_i|z_i \sim f(z_i) \qquad i = 1, \ldots, n_\cdot. \tag{3.4.19}$$

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

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

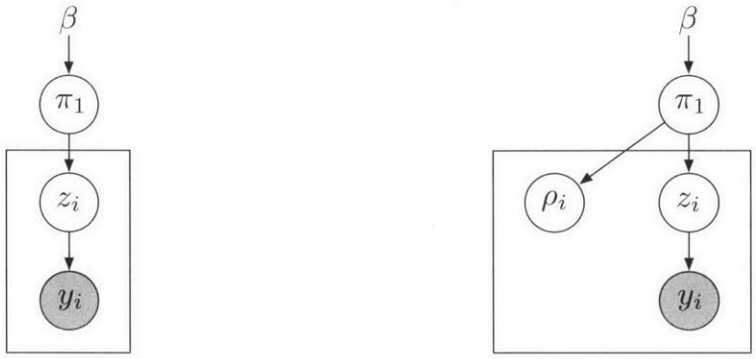



**Figure 3.6:** Simplified depiction of the relationship between the auxiliary variables and the rest of the model; 3.6(a) depicts the nonconjugate setting and 3.6(b) shows the introduced auxiliary variables $\{\rho_i\}$.

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

$$p(\pi^{(1)} | \{z_i\}, \{\rho_i\}) \tag{3.4.20}$$

$$\propto p(\pi^{(1)}) p(\{z_i\} | \pi^{(1)}) p(\{\rho_i\} | \{\pi_i^{(1)}\}) \tag{3.4.21}$$

$$\propto (\pi_1^{(1)})^{\beta_1 - 1} \cdots (\pi_L^{(1)})^{\beta_L - 1} \left( \frac{\pi_2^{(1)}}{1 - \pi_1^{(1)}} \right)^{n_2} \cdots \left( \frac{\pi_L^{(1)}}{1 - \pi_1^{(1)}} \right)^{n_L} \left( \prod_{i=1}^{n.} (\pi_1^{(1)})^{\rho_i} (1 - \pi_1^{(1)}) \right) \tag{3.4.22}$$

$$= (\pi_1^{(1)})^{\beta_1 + \sum_i \rho_i - 1} (\pi_2^{(1)})^{\beta_2 + n_2 - 1} \cdots (\pi_L^{(1)})^{\beta_L + n_L - 1} \tag{3.4.23}$$

$$\propto \text{Dir} \left( \beta_1 + \sum_i \rho_i, \beta_2 + n_2, \ldots, \beta_L + n_L \right). \tag{3.4.24}$$

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

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

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

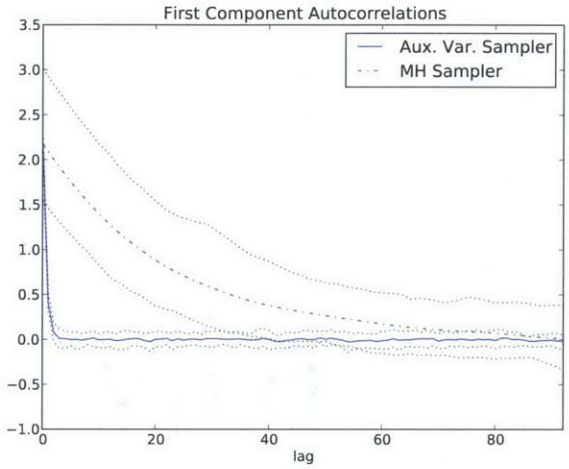
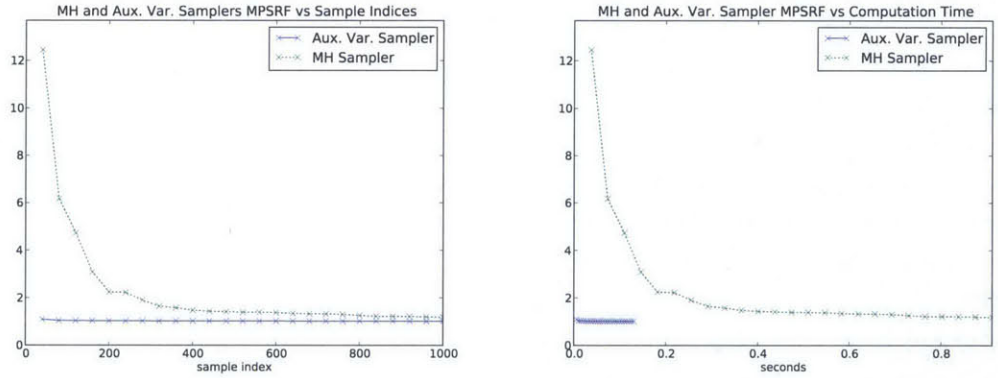



**Figure 3.7:** Empirical sample chain autocorrelation for the first component of $\pi$ for both the proposed auxiliary variable sampler and a Metropolis-Hastings sampler. The figure shows mean autocorrelations over 50 randomly-initialized runs for each sampler, with the corresponding dotted lines showing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




(a) (b)

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

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

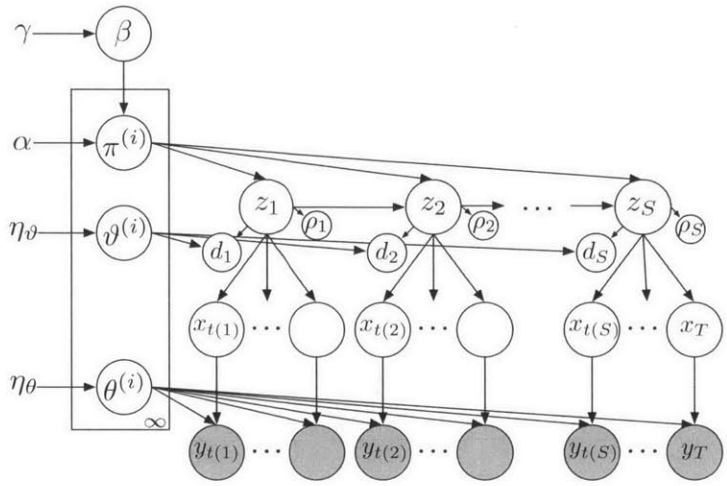



**Figure 3.9:** Graphical model for the weak-limit approximation including auxiliary variables.

### ■ 3.4.3 A Direct Assignment Sampler for the HDP-HSMM

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

Our DA sampler additionally represents the auxiliary variables necessary to recover HDP conjugacy (as introduced in the previous section). Note that the requirement for, and correctness of, the auxiliary variables described in the finite setting in Section 3.4.2 immediately extends to the infinite setting as a consequence of the Dirichlet Process's definition in terms of the finite Dirichlet distribution and the Kolmogorov extension theorem [19, Chapter 4]; for a detailed discussion, see Orbanz [87]. The connection to the finite case can also be seen in the sampling steps of the direct assignment sampler for the HDP-HMM, in which the global weights $\beta$ over $K$ instantiated components are resampled according to $(\beta_{1:K}, \beta_{\text{rest}})|\alpha, (x_t) \sim \text{Dir}(\alpha + n_1, \ldots, \alpha + n_K, \alpha)$ where $n_i$ is the number of transitions into state $i$ and Dir is the finite Dirichlet distribution.

**Resampling $(x_t)$**

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

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

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

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

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

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

To derive an expression for resampling $x_t$ in the case of the HDP-HSMM, we can similarly consider writing out an expression for the joint probability $p((x_t), (y_t)|\beta, H, G)$. However, note that as the label assignment of $x_t = k$ varies, the terms in the expression change according to whether $x_{t-1} = k$ or $x_{t+1} = k$. That is, if $x_{t-1} = k$ or $x_{t+1} = k$,

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

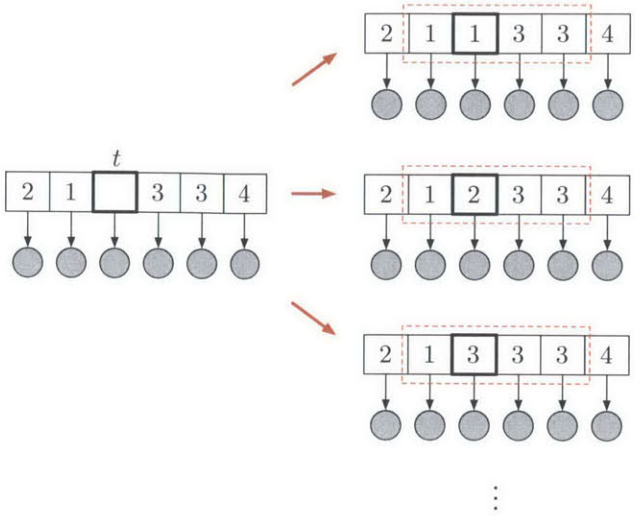



**Figure 3.10:** Illustration of the Gibbs step to resample $x_t$ for the DA sampler for the HDP-HSMM. The red dashed boxes indicate the elements of the label sequence that contribute to the score computation for $k = 1, 2, 3$ which produce two, three, and two segment terms, respectively. The label sequence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

the probability expression includes a segment term for entire contiguous run of label $k$. Hence, since we can only cancel terms that do not depend on $k$, our score expression must include terms for the adjacent segments into which $x_t$ may merge. See Figure 3.10 for an illustration.

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

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

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

The function $f_{\text{dur}}(d|k)$ is the corresponding duration predictive likelihood evaluated on a duration $d$, which depends on the durations of other segments with label $k$ and any duration hyperparameters. The function $f_{\text{obs}}$ now represents a *block* or *joint* predictive likelihood over all the data in a segment (see, for example, Murphy [79] for a thorough

discussion of the Gaussian case). Note that the denominators in the transition terms are affected by the elimination of self-transitions by a rescaling of the "total mass." The resulting chain is ergodic if the duration predictive score $f_{\text{dur}}$ has a support that includes $\{1, 2, \ldots, d_{\max}\}$, so that segments can be split and merged in any combination.

#### Resampling $\beta$ and Auxiliary Variables $\rho$

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

$$\pi_i^{(i)} | \alpha, \beta \sim \text{Beta}(\alpha\beta_i, \alpha(1-\beta_i)), \tag{3.4.29}$$

$$\rho_s | \pi_i^{(i)}, z_s \sim \text{Geo}(1 - \pi_{z_s}^{(z_s)}) \tag{3.4.30}$$

The count $n_{i,i}$, which is used in resampling the auxiliary variables $m$ of Teh et al. [106] which in turn are then used to resample $\beta$, is the total of the auxiliary variables for other segments with the same label: $n_{i,i} = \sum_{s \neq s, z_{\bar{s}} = z_s} \rho_{\bar{s}}$. This formula can be interpreted as simply sampling the number of self-transitions we may have seen at segment $s$ given $\beta$ and the counts of self- and non-self transitions in the super-state sequence. Note the diagonal entries $\pi_i^{(i)}$ are independent of the data given $(z_s)$; as before, this auxiliary variable procedure is a convenient way to integrate out numerically the diagonal entries of the transition matrix.

### ■ 3.4.4 Exploiting Changepoint Side-Information

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

In the case of the weak-limit sampler, the complexity of the bottleneck message-passing step is reduced to a function of the number of possible changepoints (instead of total sequence length): the asymptotic complexity becomes $\mathcal{O}(T_{\text{change}}^2 N + N^2 T_{\text{change}})$ ,

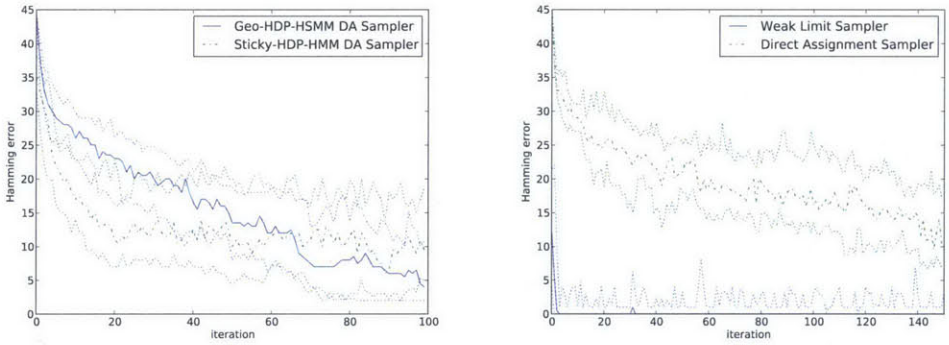
where $T_{\text{change}}$, the number of possible changepoints, may be dramatically smaller than the sequence length $T$. We simply modify the backward message-passing procedure to sum only over the possible durations:

$$\begin{aligned} B_t^*(i) &\triangleq p(y_{t+1:T}|x_{t+1} = i, x_t \neq x_{t+1}) \\ &= \sum_{d \in \mathbb{D}} B_{t+d}(i) \underbrace{\tilde{p}(D_{t+1} = d|x_{t+1} = i)}_{\text{duration prior term}} \cdot \underbrace{p(y_{t+1:t+d}|x_{t+1} = i, D_{t+1} = d)}_{\text{likelihood term}} \\ &\quad + \underbrace{\tilde{p}(D_{t+1} > T - t|x_{t+1} = i) p(y_{t+1:T}|x_{t+1} = i, D_{t+1} > T - t)}_{\text{censoring term}}, \end{aligned} \tag{3.4.31}$$

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

## ■ 3.5 Experiments

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



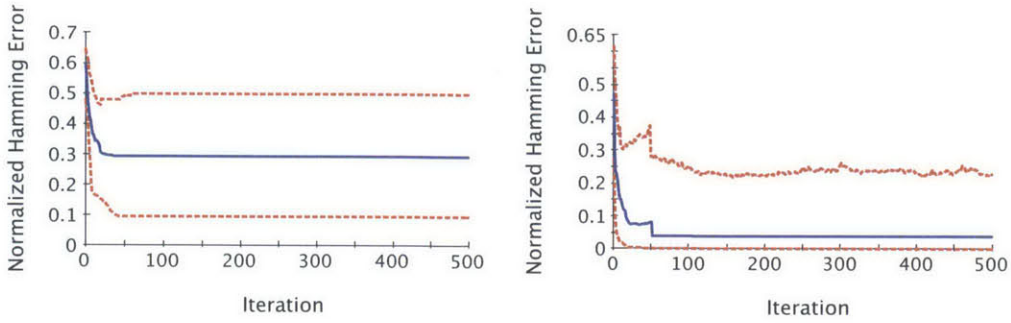

(a) (b)

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

### ■ 3.5.1 Synthetic Data

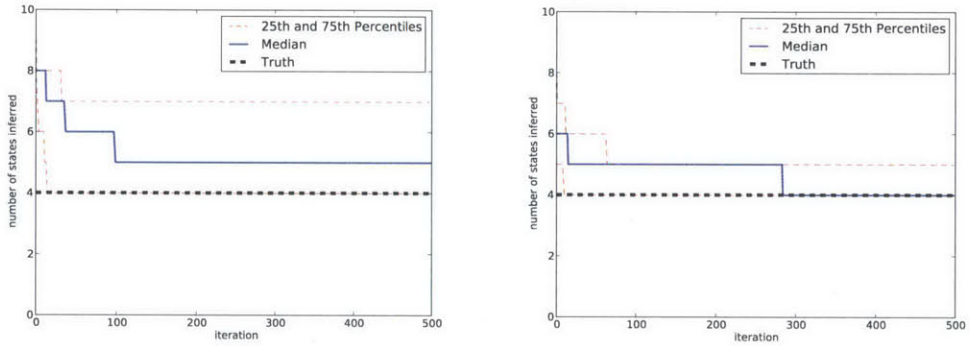
Figure 3.11 compares the HDP-HSMM direct assignment sampler to that of the Sticky HDP-HMM as well as the HDP-HSMM weak limit sampler. Figure 3.11(a) shows that the direct assignment sampler for an HDP-HSMM with geometric durations performs similarly to the Sticky HDP-HSMM direct assignment sampler when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

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




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

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

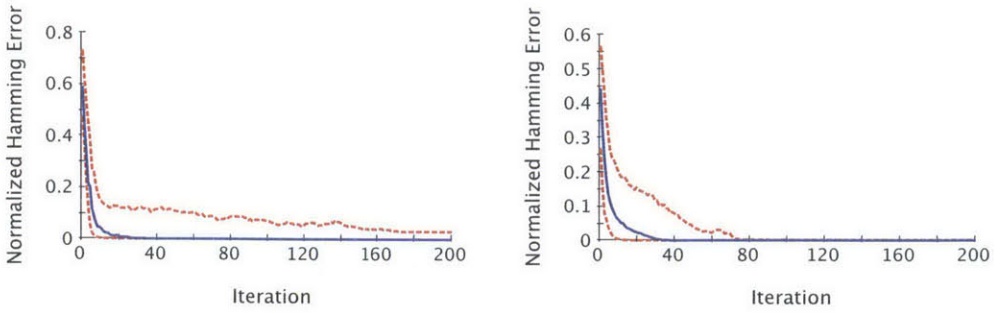



(a) HDP-HMM (b) HDP-HSMM

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

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

By setting the class of duration distributions to be a superclass of the class of geometric distributions, we can allow an HDP-HSMM model to learn an HMM from data when appropriate. One such distribution class is the class of negative binomial distributions, denoted $\text{NegBin}(r, p)$, the discrete analog of the Gamma distribution, which




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

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

covers the class of geometric distributions when $r = 1$. (We discuss negative binomial duration models further and develop specialized algorithms in Chapter 5.) By placing a (non-conjugate) prior over $r$ that includes $r = 1$ in its support, we allow the model to learn geometric durations as well as significantly non-geometric distributions with modes away from zero. Figure 3.14 shows a negative binomial HDP-HSMM learning an HMM model from data generated from an HMM with four states. The observation distribution for each state is a 10-dimensional Gaussian, with parameters sampled i.i.d. from a Normal-Inverse-Wishart (NIW) prior, as defined in Example 2.2.3 of Section 2.2. The prior over $r$ was set to be uniform on $\{1, 2, \ldots, 6\}$, and all other priors were chosen to be similarly non-informative. The sampler chains quickly concentrated at $r = 1$ for all state duration distributions. There is only a slight loss in mixing time for the HDP-HSMM compared to the HDP-HMM. This experiment demonstrates that with the appropriate choice of duration distribution the HDP-HSMM can effectively learn an HMM model.

### ■ 3.5.2 Power Disaggregation

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

of including duration information in priors as well as the significant speedup achieved with changepoint-based inference.

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

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

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

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

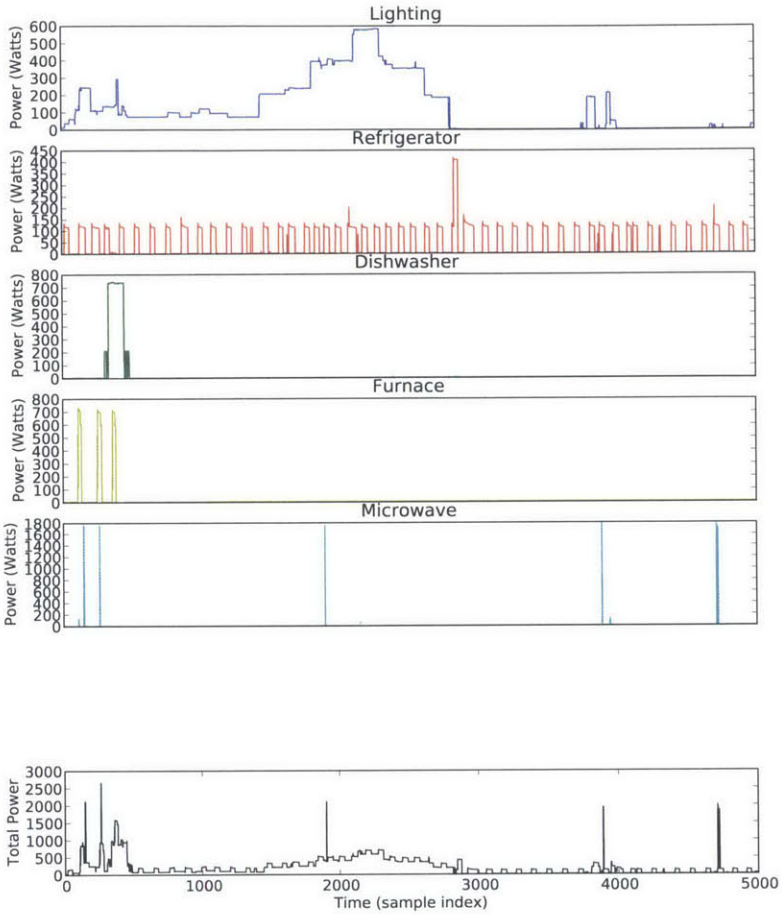



**Figure 3.15:** Example real data observation sequences for the power disaggregation experiments.

for the refrigerator device we set $L = 6$ and for lighting we set $L = 20$. We performed sampling inference independently on each observation sequence.

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

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

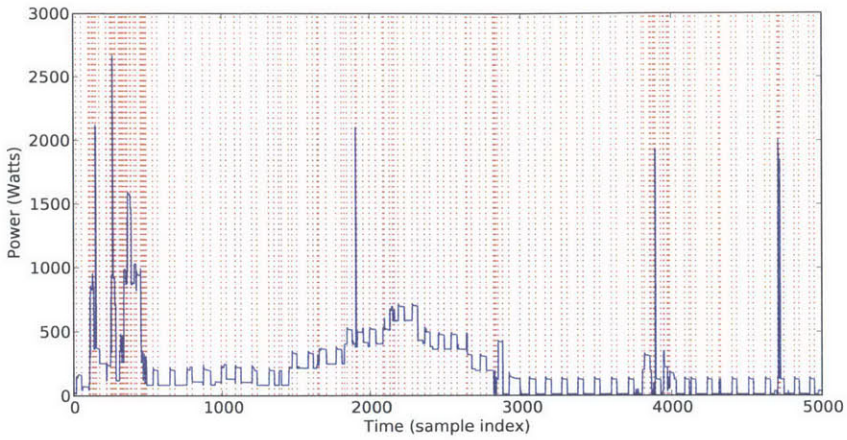



**Figure 3.16:** An total power observation sequence from the power disaggregation data set. Vertical dotted red lines indicate changepoints detected with a simple first-differences. By using the changepoint-based algorithms described in Section 3.4.4 we can greatly accelerate inference speed for this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

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

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

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

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

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

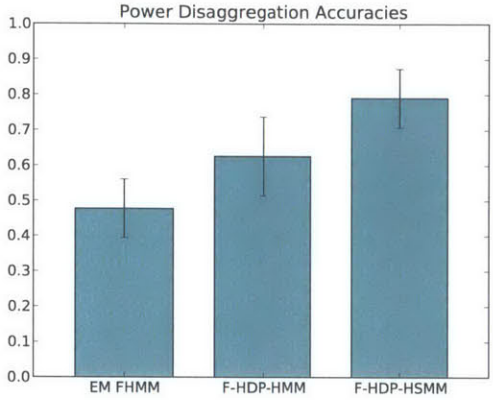



**Figure 3.17:** Overall accuracy comparison between the EM-trained FHMM of Kolter and Johnson [66], the factorial sticky HDP-HMM, and the factorial HDP-HSMM.

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

**Table 3.2:** Performance comparison broken down by house.

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

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

## ■ 3.6 Summary

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

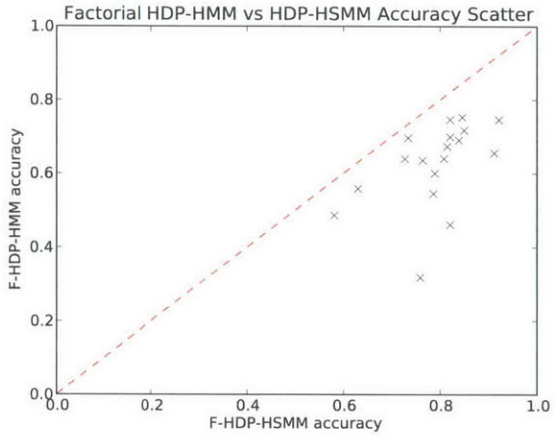



**Figure 3.18:** Performance comparison between the HDP-HMM and HDP-HSMM approaches broken down by data sequence.

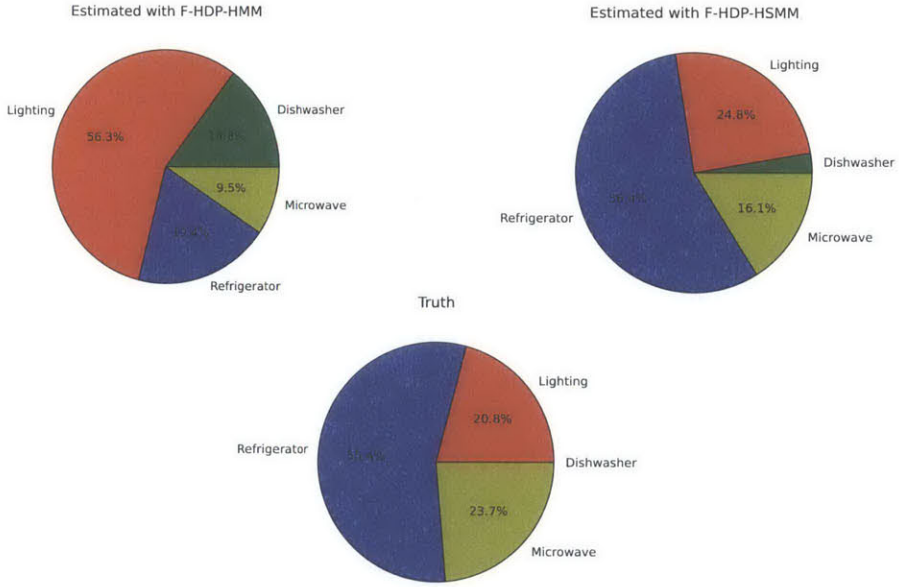



**Figure 3.19:** Estimated total power consumption for a data sequence where the HDP-HSMM significantly outperformed the HDP-HMM due to its modeling of duration regularities.

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

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

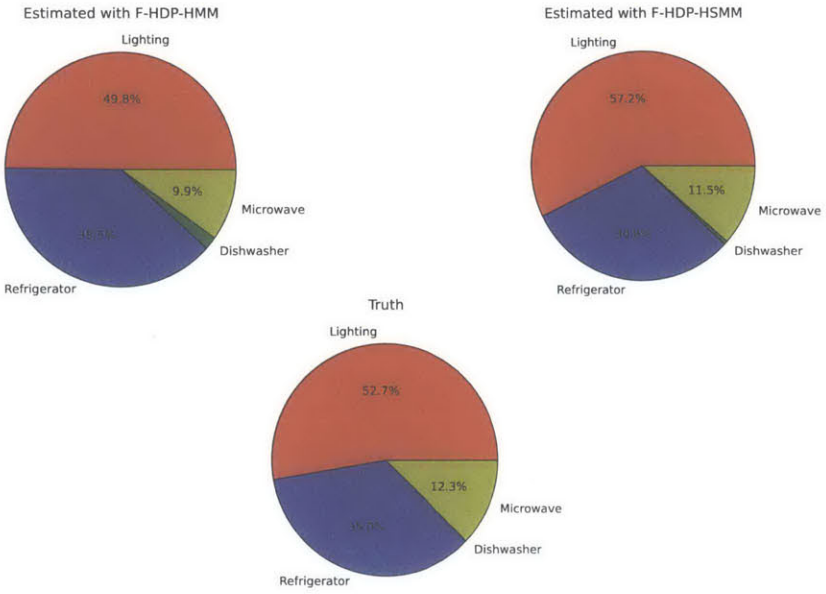



**Figure 3.20:** Estimated total power consumption for a data sequence where both the HDP-HMM and HDP-HSMM approaches performed well.

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

Chapter 4

# Faster HSMM Inference with Efficient Representations

## 4.1 Introduction

In this chapter we address some fundamental scaling challenges for Hidden semi-Markov Model (HSMM) inference. One reason HSMMs are not utilized nearly as often as the ubiquitous Hidden Markov Model (HMM) is that the HSMM's basic inference routines are often too computationally expensive in practical settings. The time complexity of HSMM inference scales quadratically with data sequence length while HMM inference scales only linearly. The slowdown relative to HMM inference is due to the weaker Markovianity properties in the HSMM state sequence, which require more expensive message passing algorithms. For this reason, with growing dataset sizes and richer hierarchical models requiring more data to be fit, HSMMs are often rejected even when explicit durations provide a better model.

We address this challenge in two ways. First, we study HSMMs with a particular natural family of duration distributions, the two-parameter negative binomial family, and develop a message passing algorithm for these models with complexity that scales only linearly with the sequence length. We derive the message passing algorithm in terms of a general notion of HMM embeddings, which we define in Section 4.3 and which improves upon previous work on expanded-state HMMs, as we discuss in Section 4.2. We also develop a Gibbs sampling algorithm that uses this linear-time message passing scheme. These algorithms immediately generalize to duration models that are mixtures of negative binomials, a natural duration analog of the Gaussian mixture model. Second, we give a linear time-invariant (LTI) system realization perspective that both generalizes the class of duration models for which HSMM message passing can be made efficient and illuminates the limits of such an approach.

In subsequent chapters we build on these inference methods for HSMMs with negative binomial durations to develop scalable inference algorithms for hierarchical Bayesian and Bayesian nonparametric models, including the stochastic variational inference (SVI)

methods for the HSMM and HDP-HSMM described in Chapter 5 and both Gibbs sampling and SVI algorithms for a new model described in Chapter 6.

The remainder of this chapter is organized as follows. In Section 4.2 we discuss previous work and how it relates to the framework and methods we develop. In Section 4.3, we provide our definition of HMM embeddings of HSMMs and give a general result on computing HSMM messages in terms of the embedding. In Section 4.4 we give a particular embedding for HSMMs with negative binomial durations and show how to use the embedding to construct an HSMM Gibbs sampling algorithm for which the time complexity of each iteration scales only linearly in the observation sequence length. Finally, in Section 4.5 we give a more general perspective on efficient representations for HSMM message passing in terms of LTI system realization.

## ■ 4.2 Related work

The work that is most closely related to the ideas we develop in this chapter is the work on expanded state HMMs (ESHMMs) [98, 97, 61, 48]. In the ESHMM framework, non-geometric state durations are captured by constructing an HMM in which several states share each observation distribution; the observations are generated from the same observation distribution while the hidden Markov chain occupies any state in the corresponding group of states, and so the effective duration distribution is modeled by the Markov dwell time for the group of states. This technique is similar to the notion of HMM embedding that we develop, but there are some key advantages to our approach, both modeling and algorithmic. First, while an ESHMM is limited to a single HMM of fixed size and transition topology, our definition of HMM embeddings as a way to compute HSMM messages allows us to perform Bayesian inference over the HSMM duration parameters and thus effectively resize the corresponding HMM embedding, as we show in Section 4.4.2. Second, another consequence of using HMM embeddings to compute HSMM messages is that the message passing recursions we develop require significantly less memory than those for the ESHMM, as we describe in Section 4.3. Finally, as we show in Section 4.5.2, our definition can be generalized to give efficient HSMM message passing algorithms for duration models that cannot be captured by ESHMMs.

Note that the HMM embedding we develop in Section 4.4 is most similar to the ESHMM Type A model [97, 61], which can model a modified negative binomial duration distribution with fixed $r$ parameter and PMF given by

$$p(k|r,p) = \binom{k-1}{k-r}(1-p)^r p^{k-r} \qquad k = r, r+1, r+2, \ldots. \tag{4.2.1}$$

This distribution is not the standard negative binomial distribution which we use in Section 4.4. In particular, the PMF (4.2.1) has support starting at $r$, which can be an artificial limitation when modeling duration distributions and particularly when learning the $r$ parameter, as we do in Section 4.4.

## ■ 4.3 HMM embeddings and HSMM messages

In this section, we give our definition for HMM embeddings of HSMMs and show how the HSMM messages can be computed in terms of HMM messages in the embeddings. In Section 4.4 we use these results to derive efficient inference algorithms for HSMMs with negative binomial durations, and in Section 4.5 we generalize these definitions.

As described in Chapter 3, there are standard HSMM forward and backward messages [78] analogous to those for the HMM. The forward messages $(F, F^*)$ and backward messages $(B, B^*)$ are defined by

$$\begin{aligned} F_{t,i} &\triangleq p(y_{1:t}, x_t = i, x_t \neq x_{t+1}) \\ &= \sum_{d=1}^{T-t-1} F^*_{t-d,i} p(d|x_{t-d} = i) p(y_{t-d:t}|x_{t-d:t} = i) \end{aligned} \tag{4.3.1}$$

$$\begin{aligned} F^*_{t,i} &\triangleq p(y_{1:t}, x_{t+1} = i|x_t \neq x_{t+1}) \\ &= \sum_{j=1}^{N} F_{t,i} p(x_{t+1} = i|x_t = j, x_t \neq x_{t+1}) \end{aligned} \tag{4.3.2}$$

$$F_{1,i} \triangleq p(x_1 = i) \tag{4.3.3}$$

$$\begin{aligned} B_{t,i} &\triangleq p(y_{t+1:T}|x_t = i, x_t \neq x_{t+1}) \\ &= \sum_{j=1}^{N} B^*_{t,j} p(x_{t+1} = j|x_t = i, x_t \neq x_{t+1}), \end{aligned} \tag{4.3.4}$$

$$\begin{aligned} B^*_{t,i} &\triangleq p(y_{t+1:T}|x_{t+1} = i, x_t \neq x_{t+1}) \\ &= \sum_{d=1}^{T-t} B_{t+d,i} p(d|x_{t+1} = i) p(y_{t+1:t+d}|x_{t+1:t+d} = i) \\ &\quad + \sum_{d=T-t+1}^{\infty} p(d|x_{t+1} = i) p(y_{t+1:T}|x_{t+1:T} = i), \end{aligned} \tag{4.3.5}$$

$$B_{T,i} \triangleq 1, \tag{4.3.6}$$

where $T$ denotes the length of the observation sequence and $N$ the number of states. These HSMM messages are expensive to compute because the expression for $B^*_{t,i}$, like that for $F_{t,i}$, involves summing over all possible durations, resulting in an overall time complexity of $\mathcal{O}(TN^2 + T^2N)$ compared to the HMM message passing complexity of $\mathcal{O}(TN^2)$. The HSMM algorithm's quadratic dependence on the sequence length $T$ severely limits the settings in which HSMM inference can be performed and has led many practitioners to prefer HMMs even when geometric state durations are unnatural.

One way to interpret the HSMM messages messages is to *embed* the HSMM into a much larger HMM that encodes the same generative process [78, 54]. The HMM embedding includes the duration information in its Markov state. Here we give a more general definition of HMM embeddings than considered in previous work, and use this general definition to explore opportunities for more efficient representations.

Recall from Chapter 3 that we parameterize an HSMM on $N$ states with a $N \times N$ transition matrix $A$ where $A_{ij} = p(x_{t+1} = j | x_t = i, x_t \neq x_{t+1})$, initial state distribution $\pi^{(0)}$, observation parameters $\theta = \{\theta^{(i)}\}_{i=1}^N$, and duration parameters $\vartheta = \{\vartheta^{(i)}\}_{i=1}^N$.

**Definition 4.3.1** (HMM embedding of an HSMM). *Given an HSMM on $N$ states with parameters $(A, \theta, \pi^{(0)}, \vartheta)$ and an observation sequence length $T$, an* HMM embedding *of the HSMM is an HMM on $\bar{N} = \sum_{i=1}^N \bar{N}^{(i)}$ states for some $\bar{N}^{(i)}$ with parameters $(\bar{A}, \bar{\theta}, \bar{\pi}^{(0)})$ that satisfy the following requirements:*

*(1) The HMM transition matrix is of the form*

$$\bar{A} = \begin{pmatrix} \bar{A}^{(1)} & & & \\ & \bar{A}^{(2)} & & \\ & & \ddots & \\ & & & \bar{A}^{(3)} \end{pmatrix} + \begin{pmatrix} | & & & \\ \bar{b}^{(1)} & & & \\ | & & & \\ & \ddots & & \\ & & | & \\ & & \bar{b}^{(N)} & \\ & & | & \end{pmatrix} \begin{pmatrix} & & \\ & A & \\ & & \end{pmatrix} \begin{pmatrix} - \bar{c}^{(1)\mathrm{T}} - & & \\ & \ddots & \\ & & - \bar{c}^{(N)\mathrm{T}} - \end{pmatrix} \tag{4.3.7}$$

*for some nonnegative matrices $\bar{A}^{(i)}$ of size $\bar{N}^{(i)} \times \bar{N}^{(i)}$ and nonnegative vectors $\bar{b}^{(i)}$ and $\bar{c}^{(i)}$ of length $\bar{N}^{(i)}$ for $i = 1, 2, \ldots, N$. Entries shown as blank are zero.*

*(2) Indexing each HMM state as $(i, j)$ for $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, \bar{N}^{(i)}$, the HMM observation parameters are $\bar{\theta}^{(i,j)} = \theta^{(i)}$ and the initial state distribution is $\bar{\pi}^{(0)}_{(i,j)} = \pi^{(0)}_i c^{(i)}_j$. We can thus associate each HSMM state $i$ with a collection of $\bar{N}^{(i)}$ HMM states of the form $(i, j)$, and we refer to each $i$ as the* state index *and each $j$ as the* pseudostate index.

*(3) The probability the HSMM assigns to any label sequence $x_{1:T} = (x_t)_{t=1}^T$ is equal to the total probability the HMM embedding assigns to all HMM state sequences*

*of the form* $\bar{x}_{1:T} = ((x_t, e_t))_{t=1}^T$ *for some sequence of pseudostate indices* $(e_t)$*; that is, writing* $p(x_{1:T}|A, \pi^{(0)}, \theta, \vartheta)$ *for the probability the HSMM assigns to a fixed label sequence* $x_{1:T}$ *and* $p(\bar{x}_{1:T}|\bar{A}, \bar{\pi}^{(0)}, \bar{\theta})$ *for the probability that the HMM assigns to a state sequence* $\bar{x}_{1:T}$ *with* $\bar{x}_t = (x_t, e_t)$*, we require*

$$p(x_{1:T}|A, \pi^{(0)}, \theta, \vartheta) = \sum_{e_t} p(\bar{x}_{1:T}|\bar{A}, \bar{\pi}, \bar{\theta}) = \sum_{e_t} p((x_t, e_t)_{t=1}^T|\bar{A}, \bar{\pi}, \bar{\theta}) \tag{4.3.8}$$

*where the sum is over all possible pseudostate index sequences.*

Note that by the construction in (4.3.7), for any HMM embedding of an HSMM, each matrix

$$\begin{pmatrix} \bar{A}^{(i)} & \bar{b}^{(i)} \\ \bar{c}^{(i)\mathsf{T}} & 0 \end{pmatrix} \tag{4.3.9}$$

for $i = 1, 2, \ldots, N$ is row-stochastic. Further, again decomposing each $\bar{x}_t = (x_t, e_t)$ into an HSMM state index and a pseudostate index, note that by Definition 4.3.1 we have

$$\bar{c}_j^{(i)} = p(\bar{x}_{t+1} = (i, j)|x_{t+1} = i, x_t \neq x_{t+1}) \tag{4.3.10}$$

$$\bar{b}_j^{(i)} = p(x_t \neq x_{t+1}|\bar{x}_t = (i, j)). \tag{4.3.11}$$

Thus we refer to each $\bar{c}^{(i)}$ as the vector of *entrance probabilities* and to each $\bar{b}^{(i)}$ as the vector of *exit probabilities* for the HMM embedding pseudostates corresponding to HSMM state $i$. An HMM embedding captures the desired HSMM transition dynamics because $\sum_{k=1}^{N^{(i)}} \sum_{\ell=1}^{N^{(j)}} p(\bar{x}_{t+1} = (j, \ell)|\bar{x}_t = (i, k), x_t \neq x_{t+1}) = A_{ij}$. Finally, note that an HMM embedding models the HSMM durations in terms of the dwell times within the group of HMM states corresponding to each HSMM state, as we make clear in Proposition 4.3.1.

First, we give an example of a generic HSMM embedding from a construction given in Murphy [78] (though not in terms of our definition of HMM embeddings).

**Example 4.3.1.** *We can construct an HMM embedding for an arbitrary HSMM by using* $\bar{N} = TN$ *total states with* $\bar{N}^{(i)} = TN$ *and*

$$\bar{A}^{(i)} = \begin{pmatrix} 0 & 1 & & & \\ & 0 & 1 & & \\ & & \ddots & \ddots & \\ & & & 0 & 1 \\ & & & & 0 \end{pmatrix} \qquad \bar{b}^{(i)} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \tag{4.3.12}$$

$$\bar{c}^{(i)\mathsf{T}} = \begin{pmatrix} p(d \geq T|\vartheta^{(i)}) & p(d = T-1|\vartheta^{(i)}) & \cdots & p(d=2|\vartheta^{(i)}) & p(d=1|\vartheta^{(i)}) \end{pmatrix}. \tag{4.3.13}$$

*In this HMM embedding, when the group of states corresponding to HSMM state $i$ is entered, a duration (censored to a maximum of $T$) is sampled from the HSMM state's duration distribution encoded in $\bar{c}^{(i)}$; if a duration $d$ is sampled, the state with pseudostate index $T - d$ is entered. Then the pseudostate index serves as a counter, deterministically incrementing until reaching the maximum pseudostate index of $T$, at which point a transition occurs and a new HSMM state is sampled.*

*While the construction makes it impossible to sample a pseudostate corresponding to a duration greater than $T$, with the label sequence length fixed at $T$ it is impossible for any duration longer than $T$ to be represented in the HSMM label sequence. Thus this HMM embedding directly models the HSMM generative process for label sequences of length $T$. Note that, in a sense we make precise in Section 4.5, an embedding that depends on the sequence length $T$ corresponds to an HSMM that does not have a finite-size realization.*

Example 4.3.1 also makes clear the computational complexity of HSMM message passing and the reason it scales quadratically with $T$. The time complexity for message passing on a generic HMM with $TN$ states is $\mathcal{O}(T(TN)^2)$ because at each time one must compute a matrix-vector product with a matrix of size $(TN) \times (TN)$. However, due to the structure of $\bar{A}$ each multiplication for the HMM embedding can be done in $\mathcal{O}(N^2 + TN)$ time. Indeed, this generic HSMM embedding provides a way to derive the HSMM messages in terms of the HMM messages on the embedding.

**Proposition 4.3.1.** *Let $(A, \theta, \pi^{(0)}, \vartheta)$ be an HSMM and let $(\bar{A}, \bar{\theta}, \bar{\pi}^{(0)})$ be a candidate HMM embedding satisfying conditions (1) and (2) of Definition 4.3.1. If for each $i = 1, 2, \ldots, N$ we have*

$$\bar{c}^{(i)\mathsf{T}} \left(\bar{A}^{(i)}\right)^{d-1} \bar{b}^{(i)} = p(d|\vartheta^{(i)}) \qquad d = 1, 2, \ldots, T-1 \tag{4.3.14}$$

$$\bar{c}^{(i)\mathsf{T}} \left(\bar{A}^{(i)}\right)^{T-1} \bar{b}^{(i)} = p(d \geq T|\vartheta^{(i)}) \tag{4.3.15}$$

*then* $(\bar{A}, \bar{\theta}, \bar{\pi}^{(0)})$ *is an HMM embedding of* $(A, \theta, \pi^{(0)}, \vartheta)$.

*Proof.* Note that the matrix expression in (4.3.14) gives the probability of absorption in $d$ timesteps for a Markov chain with transition matrix and initial state distribution given by

$$\begin{pmatrix} \bar{A}^{(i)} & \bar{b}^{(i)} \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \bar{c}^{(i)\mathsf{T}} & 0 \end{pmatrix}, \tag{4.3.16}$$

respectively. Therefore, by Definition 4.3.1, the expression gives the probability of remaining in the pseudostates corresponding to HSMM state $i$ for exactly $d$ timesteps. Finally, note that, as in Example 4.3.1, the HMM embedding only needs to represent the duration distribution for durations up to the observation sequence length $T$. □

As we show in the next example, based on a construction from Hudson [54], HMM embeddings of an HSMM are not unique.

**Example 4.3.2.** *For an HSMM* $(A, \theta, \pi^{(0)}, \vartheta)$, *using Definition 4.3.1 choose*

$$\bar{A}^{(i)} = \begin{pmatrix} 0 & 1-p(d=1|\vartheta^{(i)}) & & & \\ & 0 & 1-p(d=2|d\geq 2,\vartheta^{(i)}) & & \\ & & \ddots & \ddots & \\ & & & 0 & 1-p(d=T-1|d\geq T-1,\vartheta^{(i)}) \\ & & & & 0 \end{pmatrix} \tag{4.3.17}$$

$$\bar{b}^{(i)} = \begin{pmatrix} p(d=1|\vartheta^{(i)}) \\ p(d=2|d\geq 2,\vartheta^{(i)}) \\ \vdots \\ p(d=T-1|d\geq T-1,\vartheta^{(i)}) \\ 1 \end{pmatrix} \qquad \bar{c}^{(i)\mathsf{T}} = \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \end{pmatrix}. \tag{4.3.18}$$

*This HMM embedding also uses the pseudostate index as a duration counter, but instead of counting down until the time an HSMM transition occurs, the pseudostate here counts up the time since the previous HSMM transition.*

Given any valid HMM embedding of an HSMM, we can compute the HSMM messages in terms of the HMM messages for the embedding, as we show in the following proposition.

**Proposition 4.3.2** (HSMM Messages from HMM Embedding). *Given an HMM embedding for some HSMM, let* $\bar{F}$ *and* $\bar{B}$ *denote the HMM messages for the embedding as in Eqs.* (2.4.6) *and* (2.4.7) *of Section 2.4, so that*

$$\bar{B}_{t,(i,j)} = p(y_{t+1:T}|\bar{x}_t = (i,j)) \qquad t = 1, 2, \ldots, T \tag{4.3.19}$$

$$\bar{F}_{t,(i,j)} = p(y_{1:t}, \bar{x}_t = (i,j)) \qquad i = 1, 2, \ldots, N, \quad j = 1, 2, \ldots, \bar{N}^{(i)}. \tag{4.3.20}$$

*Then the HSMM messages for each time $t = 1, 2, \ldots, T$ and each index $i = 1, 2, \ldots, N$ can be computed as*

$$F_{t+1,i} = \sum_{j=1}^{\bar{N}^{(i)}} \bar{b}_j^{(i)} \bar{F}_{t,(i,j)} p(y_{t+1}|\theta^{(i)}) \qquad F^*_{t,i} = \sum_{j=1}^{N} F_{t,i} A_{ji} \tag{4.3.21}$$

$$B^*_{t,i} = \sum_{j=1}^{\bar{N}^{(i)}} \bar{c}_j^{(i)} \bar{B}_{t+1,(i,j)} p(y_{t+1}|\theta^{(i)}) \qquad B_{t,i} = \sum_{j=1}^{N} B^*_{t,i} A_{ij}. \tag{4.3.22}$$

*Proof.* Note that the expressions for $F^*_{t,i}$ and $B_{t,i}$ are identical to those in (4.3.2) and (4.3.4), so it suffices to check the expressions for $B^*_{t,i}$ and $F_{t+1,i}$. Using (4.3.10) and (4.3.11) we have

$$\sum_{j=1}^{N^{(i)}} \bar{b}_j^{(i)} \bar{F}_{t,(i,j)} p(y_{t+1}|\theta^{(i)}) = \sum_{j=1}^{N^{(i)}} p(y_{1:t+1}, \bar{x}_t = (i,j)) p(x_t \neq x_{t+1} | \bar{x}_t = (i,j)) \tag{4.3.23}$$

$$= p(y_{1:t+1}, x_t = i, x_t \neq x_{t+1}) = F_{t+1,i} \tag{4.3.24}$$

$$\sum_{j=1}^{N^{(i)}} \bar{c}_j^{(i)} \bar{B}_{t+1,(i,j)} p(y_{t+1}|\theta^{(i)}) = \sum_{j=1}^{N^{(i)}} p(y_{t+1:T}|\bar{x}_{t+1} = (i,j)) \cdot p(\bar{x}_{t+1} = (i,j) | x_{t+1} = i, x_t \neq x_{t+1}) \tag{4.3.25}$$

$$= p(y_{t+1:T}|x_{t+1} = i, x_t \neq x_{t+1}) = B^*_{t,i} \tag{4.3.26}$$

□

Note that, since the full HMM messages do not need to be stored, the recursions in Eqs. (4.3.21) and (4.3.22) can offer memory savings relative to simply computing the HMM messages in the HMM embedding; instead, the recursions in Eqs. (4.3.21) and (4.3.22) require only $\mathcal{O}(TN + \bar{N})$ memory to compute. In the case of the embedding of Example 4.3.1, since multiplication by each $\bar{A}^{(i)}$ only performs shifting, the HSMM messages can be computed with $\mathcal{O}(TN)$ memory, and indeed in that case the computation corresponds precisely to implementing the recursions in (4.3.1)-(4.3.5). The recursions in Eqs. (4.3.21) and (4.3.22) can be computed in time $\mathcal{O}(TN\bar{N}^2_{\max} + TN^2)$, where $\bar{N}_{\max} = \max_i \bar{N}^{(i)}$.

From the HMM embedding perspective, the HSMM message passing complexity is due to generic duration distributions requiring each HSMM state to be augmented by $T$ pseudostates in the embedding. In the next section we study a particular family of duration distributions for which an HSMM can be encoded as an HMM using many fewer pseudostates.

## ■ 4.4 HSMM inference with negative binomial durations

In this section, we develop an HMM embedding for HSMMs with negative binomial duration distributions as well as a Gibbs sampler for such models. First, we develop the HMM embedding in terms of the general results of the previous section, prove its correctness, and state its message passing computational complexity. Next, we show how to use the embedding to construct an HSMM Gibbs sampling algorithm in which the overall time complexity of each iteration scales only linearly with the sequence length $T$. Finally, we show how to generalize the HMM embedding construction to include HSMMs in which the durations are mixtures of negative binomial distributions.

### ■ 4.4.1 An embedding for negative binomial durations

The negative binomial family of discrete distributions, denoted $\mathrm{NB}(r, p)$ for parameters $r > 0$ and $0 < p < 1$, is well-studied in both frequentist and Bayesian analysis [38]. Furthermore, it has been recommended as a natural model for discrete durations [26, 61] because of it can separately parameterize mean and variance and because it includes geometric durations as a special case when $r = 1$. In this section, using our formulation of HMM embeddings from Section 4.3 we show that negative binomial distributions also provide computational advantages for HSMM message passing and inference.

The negative binomial probability mass function (PMF) can be written[1]

$$p(k|r,p) = \binom{k+r-2}{k-1}(1-p)^r p^{k-1} \qquad k = 1, 2, \ldots. \tag{4.4.1}$$

When $r$ is taken to be fixed, the family of distributions over $p$ is an exponential family:

$$p(k|r,p) = h_r(k) \exp\{\eta(p) \cdot t(k) - Z_r(p)\} \tag{4.4.2}$$

with

$$h_r(k) \triangleq \binom{k+r-2}{k-1}, \qquad \eta(p) \triangleq \ln p, \tag{4.4.3}$$

$$t(k) \triangleq k - 1, \qquad Z_r(p) \triangleq r \ln(1-p). \tag{4.4.4}$$

However, when considered as a family over $(r, p)$, the negative binomial is no longer an exponential family of distributions because the log base measure $\ln h_r(k)$ has a dependence on $r$ that does not interact linearly with a statistic of the data. The

[1] While some definitions take the support of the negative binomial PMF to be $\{0, 1, 2, \ldots\}$, for duration modeling we shift the PMF to start at 1 because we do not want to include durations of 0 length.

definition of the negative binomial can be generalized to any positive real $r$ parameter by replacing the definition of $h_r(k)$ with the appropriate ratio of gamma functions, but in this chapter we restrict our attention to the case where $r$ is a positive integer. This restriction is essential to the algorithms developed here and does not substantially reduce the negative binomial's expressiveness as a duration model. For a discussion of general results on exponential families of distributions, see Section 2.2.

To construct an efficient HMM embedding, we use the fact that a negative binomial-distributed random variable can be represented as a sum of $r$ geometrically-distributed, shifted random variables:

$$x \sim \mathrm{NB}(r,p) \quad \Longleftrightarrow \quad x = 1 + \sum_{i=1}^{r} z_i \quad \text{with} \quad z_i \overset{\text{iid}}{\sim} \mathrm{ShiftedGeo}(1-p) \tag{4.4.5}$$

where $z_i \overset{\text{iid}}{\sim} \mathrm{ShiftedGeo}(1-p)$ denotes that the $z_i$ are independent and each has a geometric distribution with parameter $1-p$ shifted so that the support includes 0, i.e. the PMF of each $z_i$ is given by

$$p(z|p) = p^z(1-p) \qquad z = 0, 1, 2, \ldots. \tag{4.4.6}$$

Therefore, given an HSMM in which the duration of state $i$ is sampled from $\mathrm{NB}(r^{(i)}, p^{(i)})$, we can construct an HMM embedding by augmenting each HSMM state with $\bar{N}^{(i)} = r^{(i)}$ pseudostates and choosing

$$\bar{A}^{(i)} = \begin{pmatrix} p^{(i)} & 1-p^{(i)} & & \\ & \ddots & \ddots & \\ & & p^{(i)} & 1-p^{(i)} \\ & & & p^{(i)} \end{pmatrix} \qquad \bar{b}^{(i)} = \begin{pmatrix} \\ \\ \\ 1-p^{(i)} \end{pmatrix} \tag{4.4.7}$$

$$\bar{c}^{(i)} = \begin{pmatrix} \mathrm{Binom}(r^{(i)}-1|r^{(i)}-1, p^{(i)}) \\ \mathrm{Binom}(r^{(i)}-2|r^{(i)}-1, p^{(i)}) \\ \vdots \\ \mathrm{Binom}(0|r^{(i)}-1, p^{(i)}) \end{pmatrix} \tag{4.4.8}$$

where $\mathrm{Binom}(k|n,p)$ denotes a binomial PMF with parameters $(n,p)$ and evaluated at $k$, given by

$$\mathrm{Binom}(k|n,p) \triangleq \binom{n}{k} p^k (1-p)^{n-k}. \tag{4.4.9}$$

In the next proposition, we show that this HMM embedding encodes the correct duration distributions.

**Proposition 4.4.1.** *Using $\bar{A}^{(i)}$, $\bar{b}^{(i)}$, and $\bar{c}^{(i)}$ as in (4.4.7) and (4.4.8) in the HMM embedding construction of Definition 4.3.1 gives a valid HMM embedding of an HSMM in which the duration of state $i$ is distributed as* $\mathrm{NB}(r^{(i)}, p^{(i)})$.

*Proof.* By Proposition 4.3.1 it suffices to show that the probability of absorption after exactly $d$ timesteps in each Markov chain with transition matrix and initial state distribution given by

$$\begin{pmatrix} \bar{A}^{(i)} & \bar{b}^{(i)} \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \bar{c}^{(i)\mathrm{T}} & 0 \end{pmatrix}, \tag{4.4.10}$$

respectively, is distributed as $\mathrm{NB}(r^{(i)}, p^{(i)})$. To simplify notation, we drop the superscript $i$ for the remainder of this proof.

Writing $X_j$ to denote the random time to absorption from state $r-j$ for $j = 0, 1, \ldots, r-1$ in the Markov chain parameterized by (4.4.10) , note that by the construction of the transition matrix we can write $X_j = 1 + Y_j + X_{j-1}$ for $j = 1, 2, \ldots, r-1$ and $X_0 = 1 + Y_0$, where $Y_j \overset{\text{iid}}{\sim} \mathrm{Geo}(p)$ for $j = 0, 1, \ldots, r-1$. Therefore $X_j = 1 + j + \sum_{k=1}^{j} Y_j$, and so we can write the PMF of $X_j$ as $\binom{k-1}{k-(r-j)}(1-p)^{r-j}p^{k-(r-j)}$ for $k = r, r+1, \ldots$.

Summing over the initial state distribution we can write probability of absorption after $k$ steps as

$$\sum_{j=0}^{r-1} \binom{r-1}{r-j-1}(1-p)^j p^{r-j-1} \cdot \binom{k-1}{k-(r-j)}(1-p)^{r-j}p^{k-(r-j)} \tag{4.4.11}$$

$$= \sum_{j=0}^{r-1} \binom{r-1}{r-j-1}\binom{k-1}{k-(r-j)}(1-p)^r p^{k-1} \tag{4.4.12}$$

$$= \binom{r+k-2}{k-1}(1-p)^r p^{k-1} \tag{4.4.13}$$

where the last line follows from the Vandermonde identity [4]

$$\binom{m+n}{\ell} = \sum_{j=0}^{\ell} \binom{m}{j}\binom{n}{\ell-j}. \tag{4.4.14}$$

□

Note that we can compute matrix-vector products against each $\bar{A}^{(i)}$ in $\mathcal{O}(r^{(i)})$ time. Therefore with the HMM embedding given in (4.4.7) and (4.4.8) and Proposition 4.3.2, for HSMMs with negative binomial durations we can compute the HSMM messages in time $\mathcal{O}(TN^2 + TNR)$, where $R = \max_i r^{(i)}$. This message passing computation avoids the quadratic dependence on $T$ necessary for generic HSMM messages.

This embedding is not unique; indeed, it can be checked that another valid HMM embedding is given by choosing

$$\bar{A}^{(i)} = \begin{pmatrix} p^{(i)} & (1-p^{(i)})(1-(1-p^{(i)})^{r^{(i)}}) & & \\ & \ddots & \ddots & \\ & & p^{(i)} & (1-p^{(i)})(1-(1-p^{(i)})^2) \\ & & & p^{(i)} \end{pmatrix} \tag{4.4.15}$$

$$\bar{b}^{(i)} = \begin{pmatrix} (1-p^{(i)})^{r^{(i)}} \\ \vdots \\ (1-p^{(i)})^2 \\ 1-p^{(i)} \end{pmatrix} \qquad \bar{c}^{(i)} = \begin{pmatrix} 1 \\ \\ \\ \end{pmatrix} \tag{4.4.16}$$

As we describe in Section 4.5.2, with respect to the HSMM forward message recursions these two alternative HMM embeddings for negative binomial durations are analogous to LTI realizations in observable canonical form and controllable canonical form, respectively.

### ■ 4.4.2 Gibbs sampling

Here we show how the HMM embedding for HSMMs with negative binomial durations can be used to construct an HSMM (or weak limit HDP-HSMM) Gibbs sampler. Unlike the corresponding Gibbs sampler developed for HSMMs with arbitrary duration distributions in Chapter 3, the time complexity of each iteration of this Gibbs sampler scales only linearly in $T$, requiring $\mathcal{O}(TN^2 + TNR)$ time for each update instead of $\mathcal{O}(TN^2 + T^2N)$.

We describe the resampling steps for both the HSMM label sequence and the negative binomial parameters; the other sampling updates to the observation parameters, transition matrix, and initial state distribution are performed as in Section 3.4. We place priors of the form $p(r^{(i)}, p^{(i)}) = p(r^{(i)})p(p^{(i)})$ over the negative binomial parameters for each $1, 2, \ldots, N$. In particular, we choose the prior over each $r^{(i)}$ to be a generic distribution with finite support $\{1, 2, \ldots, r_{\max}\}$ and parameter $\nu \in \mathbb{R}_+^N$, writing the PMF as

$$p(r|\nu) \propto \exp\left\{\ln \nu^{\mathsf{T}} \mathbb{1}_r\right\} \qquad r = 1, 2, \ldots, r_{\max} \tag{4.4.17}$$

where $\mathbb{1}_r$ denotes an indicator vector of length $N$ with its $r$th entry set to 1 and its others set to 0. We place beta priors over each $p^{(i)}$ with parameters $(a, b)$, writing

$$p(p|a,b) = \text{Beta}(a,b) = \exp\{(a-1)\ln(p) + (b-1)\ln(1-p) - \ln B(a,b)\} \tag{4.4.18}$$

for $p \in (0,1)$, where $B(a,b)$ is the beta function defined by

$$B(a,b) = \int_0^1 t^{a-1}(1-t)^{b-1}\,dt. \tag{4.4.19}$$

**Resampling the label sequence $x_{1:T}$.** Using the sampled values of each $(r^{(i)}, p^{(i)})$ and the other HSMM parameters, as outlined in the previous section we can use the HMM embedding given in (4.4.7)-(4.4.8) and the message passing recursions given in Proposition 4.3.2 to compute the HSMM backward messages in time $\mathcal{O}(TN^2 + TNR)$. Using the HSMM backward messages and the forward sampling algorithm given in Section 3.4.1, we can then construct a block sample of the label sequence $x_{1:T}$ in time $\mathcal{O}(TN)$. These steps require a total of $\mathcal{O}(TN + NR)$ memory to compute.

**Resampling the negative binomial parameters $(r^{(i)}, p^{(i)})$.** To derive a sampling update for the negative binomial parameters $(r^{(i)}, p^{(i)})$ for each state $i = 1, 2, \ldots, N$ given the sampled label sequence $x_{1:T}$, we first denote the set of durations of label $i$ in the label sequence by $\{d_k\}_{k=1}^D$, where $D$ is the number of times HSMM state $i$ is entered in the label sequence and we drop the explicit index $i$ from the notation. Then the task is then to sample $(r^{(i)}, p^{(i)})$ conditioned on the $\{d_k\}$ and the hyperparameters $a$, $b$, and $\nu$.

Suppressing explicit conditioning on hyperparameters from the notation, we can generate such a sample by first sampling $r^{(i)}|\{d_k\}$ and then sampling $p^{(i)}|r^{(i)}, \{d_k\}$. We can sample $r^{(i)}|\{d_k\}$ according to the PMF

$$p(r|\{d_k\}) \propto \int p(r)p(p)p(\{d_k\}|r,p)\,dp \tag{4.4.20}$$

$$= p(r)\prod_{k=1}^{D}\binom{d_k + r - 2}{d_k - 1} \left(\frac{1}{B(a,b)}\int \exp\left\{(a + \textstyle\sum_{k=1}^D d_k)\ln(p) + (b + rD)\ln(1-p)\right\}\,dp\right) \tag{4.4.21}$$

$$= \nu_r \prod_{k=1}^{D}\binom{d_k + r - 2}{d_k - 1}\left(\frac{B\left(a + \sum_{k=1}^D d_k, b + rD\right)}{B(a,b)}\right) \tag{4.4.22}$$

for $r = 1, 2, \ldots, r_{\max}$, where we have used the fact that, for each fixed $r$, the Beta prior

on $p$ is conjugate to the negative binomial likelihood. Using each sampled value $\hat{r}^{(i)}$ of $r^{(i)}$, we can then sample $p^{(i)}$ according to

$$p(p|\{d_k\}, r = \hat{r}^{(i)}) = \text{Beta}\left(a + \sum_{i=1}^{D} d_i, b + \hat{r}^{(i)} D\right). \tag{4.4.23}$$

Thus, using Eqs. (4.4.22) and (4.4.23) we can resample the negative binomial parameters $(r^{(i)}, p^{(i)})$ for each HSMM state $i$, completing the Gibbs sampler.

Note that by resampling the negative binomial parameters $r^{(i)}$ we are effectively performing inference over the size of the HMM embedding given in Section 4.4.1, which we only use to compute the HSMM messages during the resampling step for the label sequence $x_{1:T}$. Thus, unlike the ESHMM approach discussed in Section 4.2, we can learn a much broader class of duration distributions than those corresponding to only fixed $r^{(i)}$ parameters while maintaining efficient message passing.

In Chapter 6, we extend the ideas used to derive this Gibbs sampling algorithm to construct a scalable mean field inference algorithm for which approximate updates can also be computed in time linear in the sequence length $T$.

### ■ 4.4.3 HMM embeddings for negative binomial mixtures

Here we show how to generalize the HMM embedding for negative binomial durations to an HMM embedding for durations that are mixtures of negative binomials.

Suppose that the duration distribution for HSMM state $i$ has the PMF

$$p(d|\rho, r, p) = \sum_{j=1}^{K^{(i)}} \rho^{(i,j)} \binom{r^{(i,j)} + d - 2}{d-1} (1 - p^{(i,j)})^{r^{(i,j)}} p^{(i,j)d-1} \tag{4.4.24}$$

for mixture weights $\rho^{(i,j)}$ and negative binomial parameters $r^{(i,j)}$ and $p^{(i,j)}$, with $j = 1, 2, \ldots, K^{(i)}$ where $K^{(i)}$ is the number of mixture components. Then we can choose the embedding

$$\bar{A}^{(i)} = \begin{pmatrix} \bar{A}^{(i,1)} & & & \\ & \bar{A}^{(i,2)} & & \\ & & \ddots & \\ & & & \bar{A}^{(i,K^{(i)})} \end{pmatrix} \quad \bar{b}^{(i)} = \begin{pmatrix} \bar{b}^{(i,1)} \\ \bar{b}^{(i,2)} \\ \vdots \\ \bar{b}^{(i,K^{(i)})} \end{pmatrix} \quad \bar{c}^{(i)} = \begin{pmatrix} \bar{c}^{(i,1)} \\ \bar{c}^{(i,2)} \\ \vdots \\ \bar{c}^{(i,K^{(i)})} \end{pmatrix} \tag{4.4.25}$$

where

$$\bar{A}^{(i,j)} = \begin{pmatrix} p^{(i,j)} & 1-p^{(i,j)} & & \\ & \ddots & \ddots & \\ & & p^{(i,j)} & 1-p^{(i,j)} \\ & & & p^{(i,j)} \end{pmatrix} \qquad \bar{b}^{(i,j)} = \begin{pmatrix} \\ \\ \\ 1-p^{(i,j)} \end{pmatrix}$$

$$\bar{c}^{(i,j)} = \rho^{(i,j)} \begin{pmatrix} \text{Binom}(r^{(i,j)}-1 | r^{(i,j)}-1, p^{(i,j)}) \\ \text{Binom}(r^{(i,j)}-2 | r^{(i,j)}-1, p^{(i,j)}) \\ \vdots \\ \text{Binom}(0 | r^{(i,j)}-1, p^{(i,j)}) \end{pmatrix}. \tag{4.4.26}$$

That is, we can construct an embedding for any mixture model by a simple composition of embeddings, where the entrance probabilities $\bar{c}^{(i,j)}$ are weighted by the mixture weights $\rho^{(i,j)}$.

The time complexity for message passing in this HMM embedding is $\mathcal{O}(TNR + TN^2)$ where now $R = \max_i \sum_j r^{(i,j)}$. The memory complexity is only $\mathcal{O}(TN + RN)$ because, as with the other HMM embeddings, we do not need to store the corresponding HMM messages. Gibbs sampling updates can be performed using standard methods for mixture models [11].

## ■ 4.5 Generalizations via LTI system realization

In this section we extend the notion of HMM embedding of HSMMs developed in Section 4.3 to a notion of LTI realization of HSMMs. The contributions in this section are primarily of theoretical interest, though the framework we develop here may lead to efficient HSMM message passing for a greater variety of duration models or to new approximation schemes. In addition, by making connections to LTI systems and positive realization theory, we show the limits of such methods by giving both an example of a duration distribution that can be represented efficiently as an LTI realization but not an HMM embedding as well as an example of a duration distribution that has no efficient representation of either kind.

The remainder of this section is organized as follows. In Section 4.5.1 we review basic definitions and results from LTI system realization theory. In Section 4.5.2 we develop a definition of LTI realizations of HSMMs and show some basic results, including examples that show the limitations of such an approach to HSMM message passing. We also show that HMM embeddings correspond to (normalized) positive realizations, which are more constrained than general LTI realizations.

### ■ 4.5.1 LTI systems and realizations

In this section we review some basic definitions and state some results from linear system theory [86] and positive linear system theory [8, 25]. We use these definitions and results to define LTI realizations of HSMMs in Section 4.5.2.

We consider discrete-time, single-input single-output (SISO) linear systems of the form[2]:

$$\begin{aligned} z_{t+1} &= \bar{A} z_t + \bar{b} u_t & z_0 &= 0 \\ w_t &= \bar{c}^\mathsf{T} z_t & t &= 0, 1, 2, \ldots \end{aligned} \tag{4.5.1}$$

for input signals $u$, output signals $w$, and internal state sequence $z$. We call the triple $(\bar{A}, \bar{b}, \bar{c})$ the parameters of the LTI system, where $A$ is a $K \times K$ matrix and $b$ and $c$ are vectors of length $K$ for some $0 < K < \infty$. Note that the impulse response of the system can be expressed in terms of the parameters as $\bar{c}^\mathsf{T} \bar{A}^{t-1} \bar{b}$ for $t = 1, 2, \ldots$ and the transfer function of the system is $\bar{c}^\mathsf{T}(zI - \bar{A})^{-1}\bar{b}$ for $z \in \mathbb{C}$.

**Definition 4.5.1** (LTI realization). *Given an impulse response $h_t$ with $h_0 = 0$, we say a system of the form (4.5.1) is an* LTI realization *of $h_t$ if $h_t = \bar{c}^\mathsf{T} \bar{A}^{t-1} \bar{b}$ for $t = 1, 2, \ldots$.*

**Theorem 4.5.1.** *An impulse response $h_t$ has an LTI realization of the form (4.5.1) if and only if the corresponding transfer function $H(z) = \sum_{t=1}^{\infty} h_t z^{-t}$ is a strictly proper rational function of $z$.*

**Definition 4.5.2** (Positive realization [8]). *An LTI realization is a* positive realization *if for all nonnegative input signals the output signal and internal state sequence are nonnegative, i.e. if $\forall t\ u_t \geq 0 \implies \forall t\ w_t, z_t \geq 0$, where $z_t \geq 0$ is taken entrywise.*

**Theorem 4.5.2** (Theorem 1 [8]). *A realization $(\bar{A}, \bar{b}, \bar{c})$ is a positive realization if and only if $\bar{A}, \bar{b}, \bar{c} \geq 0$ entrywise.*

As we make precise in the next subsection, HMM embeddings of HSMMs correspond to positive realizations that are also normalized, in the sense that (4.3.9) is required to be row-stochastic.

**Definition 4.5.3** (Canonical realizations). *We say an LTI system of the form (4.5.1) is in* controllable canonical form *if the parameters $(\bar{A}, \bar{b}, \bar{c})$ have the form*

[2] We use $z_t$ to denote the system's internal state and $w_t$ to denote its output to avoid confusion with the HSMM label and observation sequences $x_{1:T}$ and $y_{1:T}$, respectively.

$$\bar{A} = \begin{pmatrix} a_1 & a_2 & \cdots & a_{K-1} & a_K \\ 1 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \end{pmatrix} \qquad \bar{b} = \begin{pmatrix} 1 \\ \\ \\ \\ \end{pmatrix} \qquad \bar{c} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_{K-1} \\ c_K \end{pmatrix} \tag{4.5.2}$$

*and we say it is in* observable canonical form *if the parameters have the form*

$$\bar{A} = \begin{pmatrix} a_1 & 1 & & \\ a_2 & & 1 & \\ \vdots & & & \ddots \\ a_{K-1} & & & 1 \\ a_K & & & \end{pmatrix} \qquad \bar{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_{K-1} \\ b_K \end{pmatrix} \qquad \bar{c} = \begin{pmatrix} 1 \\ \\ \\ \\ \end{pmatrix}. \tag{4.5.3}$$

*We note that it is always possible to write LTI realizations of impulse responses corresponding to strictly proper rational transfer functions in these two canonical forms.*

## ■ 4.5.2 LTI realizations of HSMMs

To motivate the definition of LTI realizations of HSMMs, we start from the definition of HMM embeddings developed in Section 4.3 and show that we can relax the requirement that the embedding parameters $\bar{A}^{(i)}$, $\bar{b}^{(i)}$, and $\bar{c}^{(i)}$ be nonnegative and normalized. In particular, note that using the representation (4.3.7) and Proposition 4.3.2, we can write the HSMM forward messages recursion as

$$F_{t+1,i} = \sum_{j=1}^{\bar{N}^{(i)}} \bar{b}_j^{(i)} \bar{F}_{t,(i,j)} p(y_{t+1}|\theta^{(i)}) \qquad F^*_{t+1,i} = \sum_{j=1}^{N} F_{t+1,i} A_{ji} \tag{4.5.4}$$

$$\bar{F}_{t+1,(i,j)} = \sum_{k=1}^{N^{(i)}} \bar{A}_{kj}^{(i)} \bar{F}_{t,(i,j)} p(y_{t+1}|\theta^{(i)}) + \bar{c}_j^{(i)} F^*_{t+1,i} \qquad \bar{F}_{1,(i,j)} = \pi_i^{(0)} \bar{c}_j^{(i)}. \tag{4.5.5}$$

These recursions can be represented as an interconnection of linear systems as shown in Figure 4.1. The system labeled $A$ simply applies right-multiplication by the HSMM transition matrix $A$ to its input vector. The block labeled by $z^{-1}$ denotes a delay block applied to each of its $N$ inputs. Finally, the systems labeled $D^{(i,t)}$ are the single-input single-output linear *time-varying* systems defined by

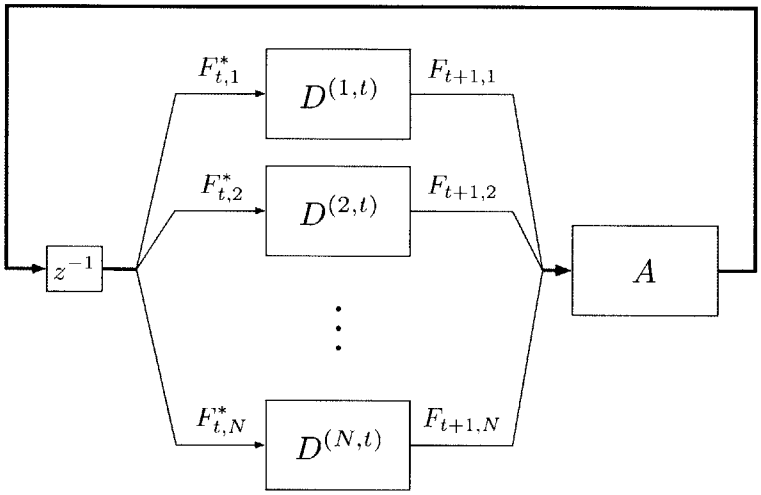



**Figure 4.1:** A block diagram showing interconnected linear systems corresponding to the forward HSMM message passing recursions in (4.5.4)-(4.5.5). The block labeled $z^{-1}$ denotes a delay block. $F_{t,i}$ and $F^*_{t,i}$ are the standard HSMM forward messages. Note that the inputs and outputs of the blocks labeled $z^{-1}$ and $A$ are vectors of length $N$, while the blocks labeled $D^{(i,t)}$ operate on each component of the vector.

$$\begin{aligned} z_{t+1} &= \bar{A}^{(i,t)} z_t + \bar{b}^{(i,t)} u_t & z_0 &= 0 \\ w_t &= \bar{c}^{(i)\mathsf{T}} z_t & t &= 0, 1, 2, \ldots . \end{aligned} \tag{4.5.6}$$

for input signal $u$ and output signal $w$, where

$$\bar{A}^{(i,t)} = p(y_t|\theta^{(i)})\bar{A}^{(i)} \qquad \bar{b}^{(i,t)} = p(y_t|\theta^{(i)})\bar{b}^{(i)}. \tag{4.5.7}$$

The corresponding recursion for the backward messages can be represented similarly.

While the linear systems defined in (4.5.6) are time-varying, as we show in the next lemma to understand when any two such systems yield the same HSMM messages it suffices to compare two corresponding LTI systems.

**Lemma 4.5.1.** *Let $(\bar{A}, \bar{b}, \bar{c})$ and $(\bar{A}', \bar{b}', \bar{c}')$ be two LTI systems and let $d_t \neq 0$ be any signal that is nonzero everywhere. The LTI systems have the same impulse response if and only if the corresponding time-varying systems of the form*

$$z_{t+1} = d_t(\bar{A} z_t + \bar{b} u_t) \qquad z'_{t+1} = d_t(\bar{A}' z_t + \bar{b}' u_t) \tag{4.5.8}$$

$$w_t = \bar{c}^{\mathsf{T}} z_t \qquad w'_t = \bar{c}'^{\mathsf{T}} z'_t, \tag{4.5.9}$$

*with $z_0 = 0$ and $z'_0 = 0$, yield the same output signal $w = w'$ given the same input signal $u$.*

*Proof.* Because the time-varying systems are linear, we can write [20] the outputs $w$ and $w'$ as linear functions of the input $u$:

$$w_t = \sum_{\ell=0}^{t-1} d(t,\ell)\bar{c}^\mathsf{T}\bar{A}^{t-\ell-1}\bar{b}u_\ell \qquad w'_t = \sum_{\ell=0}^{t-1} d(t,\ell)\bar{c}'^\mathsf{T}\bar{A}'^{t-\ell-1}\bar{b}'u_\ell \tag{4.5.10}$$

where $d(t,\ell) = \prod_{k=\ell}^{t} d_k$. If we consider inputs of the form $u_\ell = \delta(k-\ell)$ for some $k$, where $\delta(k)$ is 1 if $k$ is 0 and 0 otherwise, we see that the outputs $w_t$ and $w'_t$ are equal for all such inputs if and only if the terms in each sum are equal for each $\ell$. However, these terms are equal if and only if we have $\bar{c}^\mathsf{T}\bar{A}^{t-1}\bar{b} = \bar{c}'^\mathsf{T}\bar{A}'^{t-1}\bar{b}'$ for all $t = 1, 2, \ldots$. □

Using Lemma 4.5.1, the following proposition gives a characterization of the parameters $(\bar{A}, \bar{b}, \bar{c})$ that, when used in the recursions (4.5.4)-(4.5.5), compute the correct HSMM messages. This proposition is analogous to Proposition 4.3.1 except it does not require that the realization parameters be entrywise nonnegative or normalized, in the sense that (4.3.9) need not be row-stochastic; it only constrains the system's impulse response and not its parameters or internal state.

**Proposition 4.5.1.** *Let $(A, \theta, \pi^{(0)}, \vartheta)$ be an HSMM with $N$ states and let $(\bar{A}^{(i)}, \bar{b}^{(i)}, \bar{c}^{(i)})$ be any (not necessarily positive) linear system for each $i = 1, 2, \ldots, N$. If the impulse response of each system satisfies*

$$\bar{c}^{(i)\mathsf{T}}\left(\bar{A}^{(i)}\right)^{d-1}\bar{b}^{(i)} = p(d|\vartheta^{(i)}) \qquad d = 1, 2, \ldots, T-1, \tag{4.5.11}$$

$$\bar{c}^{(i)\mathsf{T}}\left(\bar{A}^{(i)}\right)^{T-1}\bar{b}^{(i)} = p(d \geq T|\vartheta^{(i)}) \tag{4.5.12}$$

*then using these systems in the recursions (4.5.4)-(4.5.5) yields the correct HSMM messages.*

*Proof.* If each linear system satisfies (4.5.11)-(4.5.12), then the first $T$ values of the impulse response of each system equal the first $T$ values of the impulse response for the HMM embedding of Example 4.3.1. Since the impulse responses are the same, by Lemma 4.5.1 they yield the same HSMM messages when used in the recursions (4.5.4)-(4.5.5). Therefore the linear systems $(\bar{A}^{(i)}, \bar{b}^{(i)}, \bar{c}^{(i)})$ yield the correct HSMM messages. □

Proposition 4.5.1 motivates the following definition of an LTI realization of an HSMM, which is strictly more general than the definition of HMM embedding be-

cause it allows for parameters $(\bar{A}^{(i)}, \bar{b}^{(i)}, \bar{c}^{(i)})$ that are not necessarily nonnegative and normalized and hence cannot necessarily be interpreted as encoding HMM transition probabilities.

**Definition 4.5.4** (LTI realization of an HSMM). *Given an HSMM $(A, \theta, \pi^{(0)}, \vartheta)$ on $N$ states and an observation sequence length $T$, an* LTI realization *of the HSMM is a set of $N$ LTI systems of the form 4.5.1 with parameters $(\bar{A}^{(i)}, \bar{b}^{(i)}, \bar{c}^{(i)})$ for $i = 1, 2, \ldots, N$ such that*

$$\bar{c}^{(i)\mathsf{T}} \bar{A}^{(i)d-1} \bar{b}^{(i)} = p(d|\vartheta^{(i)}) \qquad d = 1, 2, \ldots, T-1, \tag{4.5.13}$$

$$\bar{c}^{(i)\mathsf{T}} \left(\bar{A}^{(i)}\right)^{T-1} \bar{b}^{(i)} = p(d \geq T|\vartheta^{(i)}). \tag{4.5.14}$$

This definition generalizes the class of representations which can yield efficient HSMM message passing and provides connections to LTI system realization theory. In particular, it makes clear that HMM embeddings correspond to positive realizations of the duration PMF (which are also normalized in the sense that (4.3.9) must also be row-stochastic), for which the internal system state is required to remain nonnegative (and real) for all nonnegative inputs. Definition 4.5.4 removes this requirement of internal nonnegativity, and thus broadens the scope of such efficient representations from positive realizations to general LTI realizations. In particular, the internal state of an LTI system is not required to remain nonnegative or even real-valued.

While this definition is primarily of theoretical interest for the purposes of this chapter, the connection to system realization theory may allow for new algorithms and approximation schemes for HSMM message passing. There are also immediate computational advantages: by showing that the LTI system parameters need not correspond to HMM transition probabilities, it is clear that one can parameterize the system so that each matrix $\bar{A}^{(i)}$ is in bidiagonal Jordan form, and hence the overall HSMM message passing complexity reduces from $\mathcal{O}(TN^2 + TNK_{\max}^2)$ to $\mathcal{O}(TN^2 + TNK_{\max} + K_{\max}^3)$, where $K_{\max}$ denotes the size of the largest matrix $\bar{A}^{(i)}$. This bidiagonalization is not possible with HMM embeddings because HMM transition matrices may have negative and even complex-valued eigenvalues in general.

Definition 4.5.4 also leads to a natural interpretation of the alternative forms of generic HMM embeddings given in Examples 4.3.1 and 4.3.2, as well as the alternative forms of the negative binomial HMM embedding given in Section 4.4. In each of these pairs of alternative embeddings either $\bar{b}^{(i)}$ or $\bar{c}^{(i)}$ is chosen to be an indicator vector, having a single nonzero entry. While these realizations are not precisely in controllable and observable canonical forms because the structure of the corresponding $\bar{A}$ is not in canonical form, the structures of $\bar{b}^{(i)}$ and $\bar{c}^{(i)}$ are analogous to those given in

Definition 4.5.3.

We conclude this section with some examples that use the connection to LTI and positive realization theory to show the limits of both HMM embeddings and LTI realizations for constructing efficient HSMM message passing recursions.

**Example 4.5.1.** *The Poisson distribution has PMF given by* $p(d|\lambda) = \frac{\lambda^{d-1}}{(d-1)!}e^{-\lambda}$ *for a parameter* $\lambda > 0$ *and* $d = 1, 2, \ldots$. *Its probability generating function (PGF) can be written as* $\sum_{d\geq 1} p(d|\lambda)z^d = e^{-\lambda(1-z)}$. *Since its PGF is irrational, by Theorem 4.5.1 there is no finite LTI realization or HMM embedding of an HSMM with Poisson duration distributions.*[3]

**Example 4.5.2.** *Adapting Example 4 of Benvenuti and Farina [8], consider a duration distribution with PMF given by*

$$p(d|\psi) = \frac{1}{Z}(1 + \cos[(d-1)\psi])e^{-d} \qquad d \geq 1 \tag{4.5.15}$$

*for some* $\psi \in \mathbb{R}$ *and a normalization constant* $Z$. *As shown in Benvenuti and Farina [8], this duration distribution has an LTI realization with parameters*

$$A = e^{-1}\begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \bar{b} = \frac{e^{-1}}{Z}\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \qquad \bar{c} = \begin{pmatrix} 0.5 \\ 0.5 \\ 0.5 \end{pmatrix} \tag{4.5.16}$$

*but, when* $\psi/\pi$ *is irrational, it has no finite positive realization. Therefore an HSMM with such duration distributions has a finite LTI realization but no finite HMM embedding.*

## ■ 4.6 Summary

In this chapter we developed a general framework of HMM embeddings for HSMMs and showed how to compute HSMM messages using HMM embeddings. The main practical contribution, which we use in both Chapters 5 and 6, is the construction of an HMM embedding for HSMMs with negative binomial duration distributions. Using this HMM embedding, we showed how to compute HSMM messages in time that scales only linearly with the observation sequence length, and we also derived a complete Gibbs sampler for such HSMMs. The HMM embedding also generalizes to HSMMs with duration distributions that are mixtures of negative binomial distributions.

[3] While there is no finite LTI realization or HMM embedding for all possible sequence lengths $T$, the generic embedding of Example 4.3.1, in which the number of pseudostates must grow with $T$ and thus message passing complexity is quadratic in $T$, is always possible.

As a theoretical contribution, we also provided a definition of LTI realizations of HSMMs which is a strict generalization of the notion of HMM embeddings. This generalization may allow for the efficient computation of HSMM messages for a greater variety of duration distributions, and the connections to LTI realization theory may provide a basis for finding efficient approximation algorithms for message passing.

Chapter 5

# Stochastic Variational Inference for HMMs, HSMMs, and Nonparametric Extensions

Hierarchical Bayesian time series models can be applied to complex data in many domains, including data arising from behavior and motion [32, 33], home energy consumption [60], physiological signals [69], single-molecule biophysics [71], brain-machine interfaces [54], and natural language and text [44, 70]. However, for many of these applications there are very large and growing datasets, and scaling Bayesian inference in rich hierarchical models to these large datasets is a fundamental challenge.

Many Bayesian inference algorithms, including standard Gibbs sampling and mean field algorithms, require a complete pass over the data in each iteration and thus do not scale well. In contrast, some recent Bayesian inference methods require only a small number of passes [52] and can even operate in the single-pass or streaming settings [15]. In particular, stochastic variational inference (SVI) [52] provides a general framework for scalable inference based on mean field and stochastic gradient descent. However, while SVI has been studied extensively for topic models [53, 115, 17, 114, 92, 52], it has not been applied to time series.

In this chapter, we develop SVI algorithms for the core Bayesian time series models of this thesis, namely the hidden Markov model (HMM) and hidden semi-Markov model (HSMM), as well as their nonparametric extensions based on the hierarchical Dirichlet process (HDP), the HDP-HMM and HDP-HSMM. Both the HMM and HDP-HMM are ubiquitous in time series modeling, and so the SVI algorithms developed here are widely applicable. However, as discussed in the previous chapter, general HSMM inference subroutines have time complexity that scales quadratically with observation sequence length, and such quadratic scaling can be impractical even in the setting of SVI. To address this shortcoming, we use the methods developed in Chapter 4 for Bayesian inference in (HDP-)HSMMs with negative binomial durations to provide approximate

**Algorithm 5.1** Stochastic gradient ascent

Initialize $\phi^{(0)}$
**for** $t = 1, 2, \ldots$ **do**
  $\hat{k}^{(t)} \leftarrow$ sample Uniform$(\{1, 2, \ldots, K\})$
  $\phi^{(t)} \leftarrow \phi^{(t)} + \rho^{(t)} K G^{(t)} \nabla_\phi g(\phi^{(t-1)}, \bar{y}^{(\hat{k}^{(t)})})$

SVI updates with time complexity that scales only linearly with sequence length.

In Section 5.1 we briefly review the basic ingredients of SVI. In Section 5.2, we derive SVI updates for (finite) HMMs and HSMMs, and in Section 5.3 we apply the methods derived in Chapter 4 to derive faster SVI updates for HSMMs with negative binomial durations. Finally, in Section 5.4 we extend these algorithms to the nonparametric HDP-HMM and HDP-HSMM.

## ■ 5.1 Stochastic variational inference

In this section we summarize the general stochastic variational inference (SVI) framework developed in Hoffman et al. [52]. SVI involves performing stochastic gradient optimization on a mean field variational objective, so we first review basic results on stochastic gradient optimization and next provide a derivation of the form of the natural gradient of mean field objectives for complete-data conjugate models. We use the notation defined in Sections 2.3.2 and 2.4.2 throughout.

### ■ 5.1.1 Stochastic gradient optimization

Consider the optimization problem

$$\arg\max_\phi f(\phi, \bar{y}) \qquad \text{where} \qquad f(\phi, \bar{y}) = \sum_{k=1}^{K} g(\phi, \bar{y}^{(k)}) \tag{5.1.1}$$

and where $\bar{y} = \{\bar{y}^{(k)}\}_{k=1}^K$ is a fixed dataset. Using the decomposition of the objective function $f$, if $\hat{k}$ is sampled uniformly over $\{1, 2, \ldots, K\}$, we have

$$\nabla_\phi f(\phi) = K \sum_{k=1}^{K} \frac{1}{K} \nabla_\phi g(\phi, \bar{y}^{(k)}) = K \cdot \mathbb{E}_{\hat{k}} \left[ \nabla_\phi g(\phi, \bar{y}^{(\hat{k})}) \right]. \tag{5.1.2}$$

Thus we can generate approximate gradients of the objective $f$ using only one $\bar{y}^{(k)}$ at a time. A stochastic gradient ascent algorithm for a sequence of *stepsizes* $\rho^{(t)}$ and a sequence of positive definite matrices $G^{(t)}$ is given in Algorithm 5.1.

From classical results in stochastic optimization [93, 14], if the sequence of stepsizes

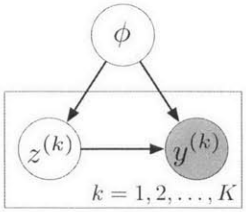



**Figure 5.1:** Prototypical graphical model for stochastic variational inference (SVI). The global latent variables are represented by $\phi$ and the local latent variables by $z^{(k)}$.

satisfies $\sum_{t=1}^{\infty} \rho^{(t)} = \infty$ and $\sum_{t=1}^{\infty} (\rho^{(t)})^2 < \infty$ and each $G^{(t)}$ has uniformly bounded eigenvalues, then the algorithm converges to a local optimum, i.e. $\phi^* \triangleq \lim_{t\to\infty} \phi^{(t)}$ satisfies $\nabla_\phi f(\phi^*, \bar{y}) = 0$ with probability 1. If $\bar{y}$ is a large dataset, then each update in a stochastic gradient algorithm only operates on one $\bar{y}^{(k)}$, or *minibatch*, at a time; therefore, stochastic gradient algorithms can scale to the large-data setting. To make a single-pass algorithm, the minibatches can be sampled without replacement. The choice of stepsize sequence can significantly affect the performance of a stochastic gradient optimization algorithm. There are automatic methods to tune or adapt the sequence of stepsizes [104, 92], though we do not discuss them here.

SVI uses a particular stochastic gradient ascent algorithm to optimize a mean field variational Bayesian objective over large datasets $\bar{y}$, as we review next.

### ■ 5.1.2 Stochastic variational inference

Using the notation of Section 2.3.2, given a probabilistic model of the form

$$p(\phi, z, y) = p(\phi) \prod_{k=1}^{K} p(z^{(k)}|\phi) p(y^{(k)}|z^{(k)}, \phi) \tag{5.1.3}$$

that includes *global* latent variables $\phi$, *local* latent variables $z = \{z^{(k)}\}_{k=1}^K$, and observations $y = \{y^{(k)}\}_{k=1}^K$, the mean field problem is to approximate the posterior $p(\phi, z|\bar{y})$ for fixed data $\bar{y}$ with a distribution of the form $q(\phi)q(z) = q(\phi) \prod_k q(z^{(k)})$ by finding a local minimum of the KL divergence from the approximating distribution to the posterior or, equivalently, finding a local maximum of the marginal likelihood lower bound

$$\mathcal{L} \triangleq \mathbb{E}_{q(\phi)q(z)} \left[ \ln \frac{p(\phi, z, \bar{y})}{q(\phi)q(z)} \right] \leq p(\bar{y}). \tag{5.1.4}$$

SVI optimizes the objective (5.1.4) using a stochastic *natural* gradient ascent algorithm over the global factors $q(\phi)$. See Figure 5.1 for a graphical model.

Gradients of $\mathcal{L}$ with respect to the parameters of $q(\phi)$ have a convenient form if we

assume the prior $p(\phi)$ and each complete-data likelihood $p(z^{(k)}, y^{(k)}|\phi)$ are a conjugate pair of exponential family densities. That is, if we have

$$\ln p(\phi) = \langle \eta_\phi, t_\phi(\phi) \rangle - Z_\phi(\eta_\phi) \tag{5.1.5}$$

$$\ln p(z^{(k)}, y^{(k)}|\phi) = \langle \eta_{zy}(\phi), t_{zy}(z^{(k)}, y^{(k)}) \rangle - Z_{zy}(\eta_{zy}(\phi)) \tag{5.1.6}$$

then conjugacy identifies the statistic of the prior with the natural parameter and log partition function of the likelihood via $t_\phi(\phi) = (\eta_{zy}(\phi), -Z_{zy}(\eta_{zy}(\phi)))$, so that

$$p(\phi|z^{(k)}, \bar{y}^{(k)}) \propto \exp\{\langle \eta_\phi + (t_{zy}(z^{(k)}, \bar{y}^{(k)}), 1), t_\phi(\phi) \rangle\}. \tag{5.1.7}$$

Conjugacy implies the optimal $q(\phi)$ has the same form as the prior; that is, without loss of generality we have $q(\phi) = \exp\{\langle \widetilde{\eta}_\phi, t_\phi(\phi) \rangle - Z_\phi(\widetilde{\eta}_\phi)\}$ for some variational parameter $\widetilde{\eta}_\phi$.

Given this structure, we can find a simple expression for the gradient of $\mathcal{L}$ with respect to the global variational parameter $\widetilde{\eta}_\phi$. To simplify notation, we write $t(z, \bar{y}) \triangleq \sum_{k=1}^K (t_{zy}(z^{(k)}, \bar{y}^{(k)}), 1)$, $\widetilde{\eta} \triangleq \widetilde{\eta}_\phi$, $\eta \triangleq \eta_\phi$, and $Z \triangleq Z_\phi$. Then we have

$$\mathcal{L} = \mathbb{E}_{q(\phi)q(z)} [\ln p(\phi|z, \bar{y}) - \ln q(\phi)] + \text{const.} \tag{5.1.8}$$

$$= \langle \eta + \mathbb{E}_{q(z)}[t(z, \bar{y})], \nabla Z(\widetilde{\eta}) \rangle - (\langle \widetilde{\eta}, \nabla Z(\widetilde{\eta}) \rangle - Z(\widetilde{\eta})) + \text{const.} \tag{5.1.9}$$

where the constant term does not depend on $\widetilde{\eta}$ and where we have used the exponential family identity $\mathbb{E}_{q(\phi)} [t_\phi(\phi)] = \nabla Z(\widetilde{\eta})$ from Proposition 2.2.2. Differentiating over $\widetilde{\eta}$, we have

$$\nabla_{\widetilde{\eta}} \mathcal{L} = \left(\nabla^2 Z(\widetilde{\eta})\right) \left(\eta + \mathbb{E}_{q(z)}[t(z, \bar{y})] - \widetilde{\eta}\right). \tag{5.1.10}$$

The factor $\nabla^2 Z(\widetilde{\eta})$ is the Fisher information of the prior $p(\phi)$ and, because the prior and variational factor are in the same exponential family, it is also the Fisher information of the global variational factor $q(\phi)$. The natural gradient $\widetilde{\nabla}_{\widetilde{\eta}}$ can be defined in terms of the gradient [52] via $\widetilde{\nabla}_{\widetilde{\eta}} \triangleq \left(\nabla^2 Z(\widetilde{\eta})\right)^{-1} \nabla_{\widetilde{\eta}}$, and so we have

$$\widetilde{\nabla}_{\widetilde{\eta}} \mathcal{L} = \left(\eta + \mathbb{E}_{q(z)}[t(z, \bar{y})] - \widetilde{\eta}\right). \tag{5.1.11}$$

Expanding $q(z) = \prod_{i=1}^K q(z^{(k)})$ and $t(z, \bar{y}) \triangleq \sum_{k=1}^K (t_{zy}(z^{(k)}, \bar{y}^{(k)}), 1)$ we can write

$$\widetilde{\nabla}_{\widetilde{\eta}} \mathcal{L} = \left(\eta + \sum_{k=1}^K \mathbb{E}_{q(z^{(k)})}[t(z^{(k)}, \bar{y}^{(k)})] - \widetilde{\eta}\right) \tag{5.1.12}$$

and so the natural gradient decomposes into local terms as required for stochastic

**Algorithm 5.2** Stochastic Variational Inference (SVI)

Initialize global variational parameter $\widetilde{\eta}_\phi^{(1)}$
**for** $t = 1, 2, \ldots$ **do**
  $\hat{k} \leftarrow$ sample Uniform($\{1, 2, \ldots, K\}$)
  $q^*(z^{(\hat{k})}) \leftarrow \textsc{LocalMeanField}(\widetilde{\eta}^{(t)}, \bar{y}^{(\hat{k})})$, e.g. Eq. (5.1.14)
  $\widetilde{\eta}_\phi^{(t+1)} \leftarrow (1-\rho^{(t)})\widetilde{\eta}_\phi^{(t)} + \rho^{(t)}\left(\eta_\phi + s \cdot \mathbb{E}_{q^*(z^{(\hat{k})})}\left[t(z^{(\hat{k})}, \bar{y}^{(\hat{k})})\right]\right)$

gradient optimization in (5.1.2).

Therefore a stochastic natural gradient ascent algorithm on the global variational parameter $\widetilde{\eta}_\phi$ proceeds at iteration $t$ by sampling a minibatch $\bar{y}^{(k)}$ and taking a step of some size $\rho^{(t)}$ in an approximate natural gradient direction via

$$\widetilde{\eta}_\phi \leftarrow (1-\rho^{(t)})\widetilde{\eta}_\phi + \rho^{(t)}\left(\eta_\phi + s \cdot \mathbb{E}_{q^*(z^{(k)})}[t(z^{(k)}, \bar{y}^{(k)})]\right) \tag{5.1.13}$$

where $q^*(x_{1:T})$ is defined below and where $s$ scales the stochastic gradient update on the minibatch to represent the full size of the dataset; that is, if $k$ is sampled uniformly and we use $|y|$ and $|y^{(k)}|$ to denote the sizes of the dataset and minibatch, respectively, we have $s = |y|/|y^{(k)}|$. In each step we find the optimal local factor $q^*(z^{(k)})$ using the standard mean field update from Proposition 2.3.3 and the current value of $q(\phi)$, i.e. we compute:

$$q^*(z^{(k)}) \propto \exp\left\{\mathbb{E}_{q(\phi)}[\ln p(z^{(k)}|\phi)p(\bar{y}^{(k)}|z^{(k)}, \phi)]\right\}. \tag{5.1.14}$$

We summarize the general SVI algorithm in Algorithm 5.2.

## ■ 5.2 SVI for HMMs and HSMMs

In this section we apply SVI to both HMMs and HSMMs and express the SVI updates in terms of HMM and HSMM messages. For notational simplicity, we consider a dataset of $K$ sequences each of length $T$, written $\bar{y} = \{\bar{y}_{1:T}^{(k)}\}_{k=1}^K$, and take each minibatch to be a single sequence written simply $\bar{y}_{1:T}$, suppressing the minibatch index $k$ for simplicity. We also assume all sequences have the same initial state distribution $\pi^{(0)}$.

### ■ 5.2.1 SVI update for HMMs

Recall from Section 2.4 that a Bayesian HMM with $N$ states defines a joint distribution over an initial state distribution $\pi^{(0)}$, a row-stochastic transition matrix $A$, observation parameters $\theta = \{\theta_i\}_{i=1}^N$, and $K$ hidden state sequences $x_{1:T}^{(k)}$ and observation sequences $y_{1:T}^{(k)}$ for $k = 1, 2, \ldots, K$. We use $\pi^{(i)}$ to denote the $i$th row of $A$ ($i = 1, 2, \ldots, N$) and $\pi = \{\pi_i\}_{i=0}^N$ to collect the transition rows and the initial state distribution. When

convenient, we use the alternative notations $p(\pi) = p(\pi^{(0)})p(A) = \prod_{i=0}^{N} p(\pi^{(i)})$ to denote the distribution over the initial state distribution and transition matrix and $p(\theta) = \prod_{i=1}^{N} p(\theta^{(i)})$ to denote the distribution over the observation parameters. The joint density for a Bayesian HMM is then

$$p(\pi^{(0)})p(A)p(\theta)\prod_{k=1}^{K} p(x_{1:T}^{(k)}, y_{1:T}^{(k)}|\pi^{(0)}, A, \theta). \tag{5.2.1}$$

In terms of the notation in Section 5.1.2, the global variables are the HMM parameters and the local variables are the hidden states; that is, $\phi = (A, \pi^{(0)}, \theta)$ and $z = x_{1:T}$. To derive explicit conjugate updates, we assume the observation model is conjugate in that $(p(\theta^{(i)}), p(y|\theta^{(i)}))$ is a conjugate pair of exponential family densities for each $i = 1, 2, \ldots, N$ and write

$$p(\pi^{(i)}) = p(\pi^{(i)}|\alpha^{(i)}) = \mathrm{Dir}(\alpha^{(i)}) \qquad i = 0, 1, \ldots, N \tag{5.2.2}$$

$$p(\theta^{(i)}) = p(\theta^{(i)}|\eta_\theta^{(i)}) = \exp\{\langle \eta_\theta^{(i)}, t_\theta^{(i)}(\theta^{(i)})\rangle - Z_\theta^{(i)}(\eta_\theta^{(i)})\} \qquad i = 1, 2, \ldots, N \tag{5.2.3}$$

$$p(y_t|\theta^{(i)}) = \exp\{\langle t_\theta^{(i)}(\theta^{(i)}), (t_y^{(i)}(y_t), 1)\rangle\} \qquad i = 1, 2, \ldots, N. \tag{5.2.4}$$

Correspondingly the variational family is $q(\pi)q(A)q(\theta)\prod_{k=1}^{K} q(x_{1:T}^{(k)})$ with

$$q(\pi^{(i)}) = q(\pi^{(i)}|\widetilde{\alpha}^{(i)}) = \mathrm{Dir}(\widetilde{\alpha}^{(i)}) \qquad i = 0, 1, \ldots, N \tag{5.2.5}$$

$$q(\theta^{(i)}) = q(\theta^{(i)}|\widetilde{\eta}_\theta^{(i)}) = \exp\{\langle \widetilde{\eta}_\theta^{(i)}, t_\theta^{(i)}(\theta^{(i)})\rangle - Z_\theta^{(i)}(\widetilde{\eta}_\theta^{(i)})\} \qquad i = 1, 2, \ldots, N. \tag{5.2.6}$$

That is, each variational factor is in the same (conjugate) prior family as the corresponding factor in the joint distribution $p$. Therefore we wish to optimize over the variational parameters for the initial state distribution $\widetilde{\alpha}^{(0)}$, the variational parameters for the transition distribution $\widetilde{\alpha}^{(i)}$ $(i = 1, 2, \ldots, N)$, and the variational parameters for the observation parameter distributions $\widetilde{\eta}_\theta$.

At each iteration of the SVI algorithm we sample a sequence $\bar{y}_{1:T}$ from the dataset and perform a stochastic gradient step on $q(A)q(\pi^{(0)})q(\theta)$ of some size $\rho$. To compute the gradient, we collect expected sufficient statistics with respect to the optimal factor for $q(x_{1:T})$, which in turn depends on the current value of $q(A)q(\pi^{(0)})q(\theta)$. Recall from Section 2.4.2 that we define

$$\widetilde{\pi}^{(i)} \triangleq \mathbb{E}_{q(\pi)}\left[\ln \pi^{(i)}\right] \qquad \widetilde{L}_{ij} \triangleq \mathbb{E}_{q(\theta)}\left[\ln p(\bar{y}_t|\theta^{(i)})\right] \tag{5.2.7}$$

**Algorithm 5.3** HMM SVI

Initialize global variational parameters $\widetilde{\eta}_\theta^{(i)}$, $\widetilde{\alpha}^{(i)}$, and $\widetilde{\alpha}^{(0)}$
**for** $t = 1, 2, \ldots$ **do**
  Sample minibatch index $\hat{k}$ uniformly from $\{1, 2, \ldots, K\}$
  Using minibatch $\bar{y}^{(\hat{k})}$, compute each $\hat{t}_y^{(i)}$, $\hat{t}_{\text{trans}}^{(i)}$, and $\hat{t}_{\text{init}}^{(i)}$
    with Eqs. (5.2.8)-(5.2.10)
  Update each $\widetilde{\eta}_\theta^{(i)}$, $\widetilde{\alpha}^{(i)}$, and $\widetilde{\alpha}^{(0)}$
    with Eqs. (5.2.11)-(5.2.13)

and collect the $\widetilde{\pi}^{(i)}$ into a matrix $\widetilde{A}$, where the $i$th row of $\widetilde{A}$ is $\widetilde{\pi}^{(i)}$. Then using the HMM messages $F$ and $B$ defined in Section 2.4 we write the expected statistics as

$$\hat{t}_y^{(i)} \triangleq \mathbb{E}_{q(x_{1:T})} \sum_{t=1}^{T} \mathbb{I}[x_t = i] t_y^{(i)}(\bar{y}_t) = \sum_{t=1}^{T} F_{t,i} B_{t,i} \cdot (t_y^{(i)}(\bar{y}_t), 1)/Z \tag{5.2.8}$$

$$(\hat{t}_{\text{trans}}^{(i)})_j \triangleq \mathbb{E}_{q(x_{1:T})} \sum_{t=1}^{T-1} \mathbb{I}[x_t = i, x_{t+1} = j] = \sum_{t=1}^{T-1} F_{t,i} \widetilde{A}_{i,j} \widetilde{L}_{t+1,j} B_{t+1,j}/Z \tag{5.2.9}$$

$$(\hat{t}_{\text{init}})_i \triangleq \mathbb{E}_{q(x_{1:T})} \mathbb{I}[x_1 = i] = \widetilde{\pi}_0 B_{1,i}/Z \tag{5.2.10}$$

where $\mathbb{I}[\,\cdot\,]$ is 1 if its argument is true and 0 otherwise and $Z$ is the normalizer $Z \triangleq \sum_{i=1}^{N} F_{T,i}$.

With these expected statistics, taking a natural gradient step in the parameters of $q(A)$, $q(\pi_0)$, and $q(\theta)$ of size $\rho$ is

$$\widetilde{\eta}_\theta^{(i)} \leftarrow (1-\rho)\widetilde{\eta}_\theta^{(i)} + \rho(\eta_\theta^{(i)} + s \cdot \hat{t}_y^{(i)}) \tag{5.2.11}$$

$$\widetilde{\alpha}^{(i)} \leftarrow (1-\rho)\widetilde{\alpha}^{(i)} + \rho(\alpha^{(i)} + s \cdot \hat{t}_{\text{trans}}^{(i)}) \tag{5.2.12}$$

$$\widetilde{\alpha}^{(0)} \leftarrow (1-\rho)\widetilde{\alpha}^{(0)} + \rho(\alpha^{(0)} + s \cdot \hat{t}_{\text{init}}^{(i)}) \tag{5.2.13}$$

where $s = |\bar{y}|/|\bar{y}^{(k)}|$ scales the minibatch gradient to represent the full dataset, as in Section 5.1. When the dataset comprises $K$ sequences where the length of sequence $k$ is $T^{(k)}$, we have $s = (\sum_{k'=1}^{K} T^{(k')})/T^{(k)}$.

We summarize the overall algorithm in 5.3.

### ■ 5.2.2 SVI update for HSMMs

The SVI updates for the HSMM are similar to those for the HMM with the addition of a duration update, though expressing the expected sufficient statistics in terms of the

HSMM messages is substantially different. The form of these expected statistics follows from the HSMM E-step [78, 54].

To derive explicit updates, we assume the duration prior and likelihood are a conjugate pair of exponential families. Writing the duration parameters as $\vartheta = \{\vartheta^{(i)}\}_{i=1}^N$, we can write the prior, variational factor, and likelihood up to proportionality as

$$p(\vartheta^{(i)}) \propto \exp\{\langle \eta_\vartheta^{(i)}, t_\vartheta^{(i)}(\vartheta^{(i)})\rangle\}, \tag{5.2.14}$$

$$p(d|\vartheta^{(i)}) = \exp\{\langle t_\vartheta^{(i)}(\vartheta^{(i)}), (t_d(d), 1)\rangle\}, \tag{5.2.15}$$

$$q(\vartheta^{(i)}) \propto \exp\{\langle \widetilde{\eta}_\vartheta^{(i)}, t_\vartheta^{(i)}(\vartheta^{(i)})\rangle\}. \tag{5.2.16}$$

Using the HSMM messages $(F, F^*)$ and $(B, B^*)$ with $\widetilde{L}$ and $\widetilde{A}$ from the previous section, we can write

$$(\hat{t}_{\text{trans}}^{(i)})_j \triangleq \mathbb{E}_{q(x_{1:T})} \sum_{t=1}^{T-1} \mathbb{1}[x_t = i, x_{t+1} = j, x_t \neq x_{t+1}] \tag{5.2.17}$$

$$= \sum_{t=1}^{T-1} F_{t,i} B_{t,j}^* \widetilde{A}_{i,j} / Z \tag{5.2.18}$$

where $Z$ is the normalizer $Z \triangleq \sum_{i=1}^N B_{0,i}^* \widetilde{\pi}_i^{(0)}$.

To be written in terms of the HSMM messages the expected label sequence indicators $\mathbb{1}[x_t = i]$ must be expanded to

$$\mathbb{1}[x_t = i] = \sum_{\tau < t} \mathbb{1}[x_{\tau+1} = i, x_\tau \neq x_{\tau+1}] - \mathbb{1}[x_\tau = i, x_\tau \neq x_{\tau+1}]. \tag{5.2.19}$$

Intuitively, this expansion expresses that a state is occupied after a transition into it occurs and until the first transition occurs out of that state and to another. Then we have

$$\mathbb{E}_{q(x_{1:T})} \mathbb{1}[x_{t+1} = i, x_t \neq x_{t+1}] = F_{t,i}^* B_{t,i}^* / Z \tag{5.2.20}$$

$$\mathbb{E}_{q(x_{1:T})} \mathbb{1}[x_t = i, x_t \neq x_{t+1}] = F_{t,i} B_{t,i} / Z. \tag{5.2.21}$$

from which we can compute $\mathbb{E}_{q(x_{1:T})} \mathbb{1}[x_t = i]$, which we use in the definition of $\hat{t}_y^{(i)}$ given in (5.2.8).

Finally, defining $\widetilde{D}_{di} \triangleq \mathbb{E}_{q(\vartheta)}\left[p(d|\vartheta^{(i)})\right]$, we compute the expected duration statistics as indicators on every possible duration $d$ via

**Algorithm 5.4** HSMM SVI

Initialize global variational parameters $\widetilde{\eta}_\vartheta^{(i)}$, $\widetilde{\eta}_\theta^{(i)}$, $\widetilde{\alpha}^{(i)}$, and $\widetilde{\alpha}^{(0)}$
**for** $t = 1, 2, \ldots$ **do**
  Sample minibatch index $\hat{k}$ uniformly from $\{1, 2, \ldots, K\}$
  Using minibatch $\bar{y}^{(\hat{k})}$, compute each $\hat{t}_{\text{dur}}^{(i)}$, $\hat{t}_y^{(i)}$, $\hat{t}_{\text{trans}}^{(i)}$, and $\hat{t}_{\text{init}}^{(i)}$
    with Eqs. (5.2.8),(5.2.10), (5.2.18), and (5.2.23)
  Update each $\widetilde{\eta}_\vartheta^{(i)}$, $\widetilde{\eta}_\theta^{(i)}$, $\widetilde{\alpha}^{(i)}$, and $\widetilde{\alpha}^{(0)}$
    with Eqs. (5.2.11)-(5.2.13) and (5.2.24)

$$(\hat{t}_{\text{dur}}^{(i)})_d \triangleq \mathbb{E}_{q(x_{1:T})}\left[\sum_t \mathbb{1}[x_t \neq x_{t+1}, x_{t+1:t+d} = i, x_{t+d+1} \neq i]\right] \tag{5.2.22}$$

$$= \sum_{t=1}^{T-d+1} \widetilde{D}_{d,i} F_{t,i}^* B_{t+d,i} \Big(\prod_{t'=t}^{t+d} \widetilde{L}_{t',i}\Big)/Z. \tag{5.2.23}$$

Note that this step alone requires $\mathcal{O}(T^2 N)$ time.

With these expected statistics, the updates to the observation, transition, and initial state factors are (5.2.11), (5.2.12), and (5.2.13). The duration factor update is

$$\widetilde{\eta}_\vartheta^{(i)} \leftarrow (1-\rho)\widetilde{\eta}_\vartheta^{(i)} + \rho\Big(\eta_\vartheta^{(i)} + s\Big(\sum_{d=1}^{T} (\hat{t}_{\text{dur}}^{(i)})_d \cdot (t_d(d), 1)\Big)\Big). \tag{5.2.24}$$

We summarize the overall algorithm in 5.4.

While these updates can be used for any family of duration models, they can be computationally expensive: as described in Chapter 4, both computing the HSMM messages and computing the expected statistics (5.2.22) require time that scales quadratically with the sequence length $T$, which can be severely limiting even in the minibatch setting. In the next section, we apply the techniques developed in Chapter 4 to the SVI algorithm to derive updates for which the computational complexity scales only linearly with $T$.

## ■ 5.3 Linear-time updates for negative binomial HSMMs

General HSMM inference is much more expensive than HMM inference, having runtime $\mathcal{O}(T^2N + TN^2)$ compared to just $\mathcal{O}(TN^2)$ on $N$ states and a sequence of length $T$. The quadratic dependence on $T$ can be severely limiting even in the minibatch setting of SVI, since minibatches often must be sufficiently large for good performance [52, 15]. In this section, we develop approximate SVI updates for a particular class of duration

distributions with unbounded support for which the computational complexity is only linear in $T$.

Following the development in Chapter 4, we consider HSMMs with negative binomial duration distributions. Each duration likelihood has parameters $r$ and $p$ with the form

$$p(k|r,p) = \binom{k+r-2}{k-1} \exp\{(k-1)\ln p + r\ln(1-p)\} \tag{5.3.1}$$

for $k = 1, 2, \ldots$. The negative binomial likelihood is not an exponential family of densities over $(r,p)$, and it has no simple conjugate prior. We use priors of the form $p(r,p) = p(r)p(p)$ with $p(r)$ a finite categorical distribution with support $\{1, 2, \ldots, r_{\max}\}$ and $p(p)$ an independent Beta distribution, i.e.

$$p(r) \propto \exp\{\langle \nu, \mathbb{1}_r \rangle\}, \quad p(p) = \mathrm{Beta}(a,b) \propto \exp\{(a-1)\ln(p) + (b-1)\ln(1-p)\}. \tag{5.3.2}$$

Similarly, we define a corresponding mean field factor $q(r,p) = q(r)q(p|r)$ as

$$q(r) \propto \exp\{\langle \widetilde{\nu}, \mathbb{1}_r \rangle\}, \qquad q(p|r) = \mathrm{Beta}(\widetilde{a}^{(r)}, \widetilde{b}^{(r)}). \tag{5.3.3}$$

Thus for $N$ states we have prior hyperparameters $\{(\nu^{(i)}, a^{(i)}, b^{(i)})\}_{i=1}^N$ and variational parameters $\{(\nu^{(i)}, \{a^{(r,i)}, b^{(r,i)}\}_{r=1}^{r_{\max}})\}_{i=1}^N$. To simplify notation, we suppress the indices $r$ and $i$ when possible.

We write $d^{(i)}(x_{1:T})$ to denote the set of durations for state $i$ in the state sequence $x_{1:T}$. Dropping indices for simplicity, the part of the variational lower bound objective that depends on $q(r,p)$ is

$$\mathcal{L} \triangleq \mathbb{E}_{q(r,p)q(x_{1:T})}\left[\ln \frac{p(r,p,d(x_{1:T}))}{q(r,p)}\right] \tag{5.3.4}$$

$$= \mathbb{E}_{q(r)} \ln \frac{p(r)}{q(r)} + \mathbb{E}_{q(r)q(x_{1:T})} h(r, d(x_{1:T})) + \mathbb{E}_{q(r)}\left\{\mathbb{E}_{q(p|r)} \ln \frac{p(p)\bar{p}(d(x_{1:T})|r,p)}{q(p|r)}\right\} \tag{5.3.5}$$

where $h(r, d(x_{1:T})) \triangleq \sum_{d' \in d(x_{1:T})} \ln \binom{r+d'-2}{d'-1}$ arises from the negative binomial base measure term and $\ln \bar{p}(d(x_{1:T})|r,p) \triangleq \sum_{d' \in d(x_{1:T})} (d' \ln p + r\ln(1-p))$ collects the negative binomial PMF terms excluding the base measure.

First, we show that the SVI updates to each $q(p|r)$ can be considered independent of each other and of $q(r)$ by taking the natural gradient of $\mathcal{L}$. The only terms in (5.3.4) that depend on $q(p|r)$ are in the final term. Since the expectation over $q(r)$ in the

final term is simply a weighted finite sum, taking the gradient of $\mathcal{L}$ with respect to the parameters $(\widetilde{a}^{(r)}, \widetilde{b}^{(r)})$ for $r = 1, 2, \ldots, r_{\max}$ yields a sum of gradients weighted by each $q(r)$. Each gradient in the sum is that of a variational lower bound with fixed $r$, and because $q(p|r)$ is conjugate to the negative binomial likelihood with fixed $r$, each gradient has a simple conjugate form. As a result of this decomposition, if we collect the variational parameters of $q(r, p)$ into $\widetilde{\eta}_\vartheta \triangleq (\widetilde{\nu}, (\widetilde{a}^{(1)}, \widetilde{b}^{(1)}), \ldots, (\widetilde{a}^{(r_{\max})}, \widetilde{b}^{(r_{\max})}))$, then the Fisher information matrix

$$J(\widetilde{\eta}_\vartheta) \triangleq \mathbb{E}_{(r,p)\sim q(r,p)} \left[ (\nabla_{\widetilde{\eta}_\vartheta} \ln q(r,p))(\nabla_{\widetilde{\eta}_\vartheta} \ln q(r,p))^\mathsf{T} \right] \tag{5.3.6}$$

is block diagonal with the same partition structure as $\widetilde{\eta}_\vartheta$. If we denote the Fisher information of $q(p|r)$ as $J(\widetilde{a}^{(r)}, \widetilde{b}^{(r)})$ , then the $(r+1)$th diagonal block of $J(\widetilde{\eta}_\vartheta)$ can be written as $q(r)J(\widetilde{a}^{(r)}, \widetilde{b}^{(r)})$, and so the $q(r)$ factors cancel in the natural gradient. Therefore the natural gradient updates to each $(\widetilde{a}^{(r)}, \widetilde{b}^{(r)})$ are independent and can be computed using simple conjugate Beta updates.

Next, we derive updates to $q(r)$. Since $q(r)$ is a discrete distribution with finite support, we write its complete-data conditional in an exponential family form trivially:

$$p(r|p, d(x_{1:T})) \propto \exp\{\langle \nu + t_r(p, d(x_{1:T})), \mathbb{1}_r \rangle\} \tag{5.3.7}$$

$$(t_r(p, d(x_{1:T})))_r \triangleq \textstyle\sum_{d' \in d(x_{1:T})} \ln p(p|d', r) + \ln h(r, d'). \tag{5.3.8}$$

From the results in Section 5.1.2 the $j$th component of the natural gradient of (5.3.4) with respect to the parameters of $q(r)$ is

$$\left(\widetilde{\nabla}_{\widetilde{\nu}} \mathcal{L}\right)_j = \nu_j + \mathbb{E}_{q(p|r=j)q(x_{1:T})} t_r(p, d(x_{1:T})) - \widetilde{\nu}_j \tag{5.3.9}$$

Due to the log base measure term $\ln h(r, d')$ in (5.3.8), these expected statistics require $\mathcal{O}(T^2 N)$ time to compute exactly even after computing the HSMM messages using (5.2.23). The HSMM SVI algorithm developed in Section 5.2.2 provides an exact algorithm using this update. However, we can use the efficient sampling-based algorithms developed in Chapter 4 to compute an approximate update more efficiently.

To achieve an update runtime that is linear in $T$, we use a sampling method inspired by the sampling-based SVI update used in Wang and Blei [114]. For some sample count $S$, we collect $S$ model parameter samples $\{(\hat{\pi}^{(\ell)}, \hat{\theta}^{(\ell)}, \hat{r}^{(\ell)}, \hat{p}^{(\ell)})\}_{\ell=1}^S$ using the current global mean field factors according to

$$\hat{\pi}^{(\ell)} \sim q(\pi) \qquad \hat{\theta}^{(\ell)} \sim q(\theta) \qquad (\hat{r}^{(\ell)}, \hat{p}^{(\ell)}) \sim q(r, p). \tag{5.3.10}$$

and for each set of parameters we sample a state sequence

$$\hat{x}_{1:T}^{(\ell)} \sim p(x_{1:T}|\bar{y}_{1:T}, \hat{\pi}^{(\ell)}, \hat{\theta}^{(\ell)}, \hat{r}^{(\ell)}, \hat{p}^{(\ell)}). \tag{5.3.11}$$

Using the methods developed in Chapter 4, each such sample can be drawn in time $\mathcal{O}(TNR + TN^2)$. We denote the set of state sequence samples as $\mathcal{S} = \{\hat{x}_{1:T}^{(\ell)}\}_{\ell=1}^{S}$ and we set $\hat{q}(x_{1:T}) = \frac{1}{S}\sum_{\hat{x}\in\mathcal{S}} \delta_{\hat{x}}(x_{1:T})$. As the number of samples $S$ grows, the distribution $\hat{q}(x_{1:T})$ approximates $\mathbb{E}_{q(\pi)q(\theta)q(r,p)}[p(x_{1:T}|\bar{y}_{1:T}, \pi, \theta, r, p)]$, while the optimal mean field update sets $q(x_{1:T}) \propto \exp\left\{\mathbb{E}_{q(\pi)q(\theta)q(r,p)} \ln p(x_{1:T}|\bar{y}_{1:T}, \pi, \theta, r, p)\right\}$. As discussed in Wang and Blei [114], since this sampling approximation does not optimize the variational lower bound directly, it should yield an inferior objective value. However, Wang and Blei [114] found this approximate SVI update yielded better predictive performance in some topic models, and provided an interpretation as an approximate expectation propagation (EP) update. As we show in Section 5.5, this update can be very effective for fitting HSMMs as well.

Given the sample-based representation $\hat{q}(x_{1:T})$, it is easy to compute the expectation over states in (5.3.9) by plugging in the sampled durations. The update to the parameters of $q(r^{(i)}, p^{(i)})$ becomes

$$\widetilde{\nu}^{(i)} \leftarrow (1-\rho)\widetilde{\nu}^{(i)} + \rho\left(\nu^{(i)} + s \cdot \hat{t}_r^{(i)}\right) \tag{5.3.12}$$

$$\widetilde{a}^{(i,r)} \leftarrow (1-\rho)\widetilde{a}^{(i,r)} + \rho\left(a^{(i)} + s \cdot \hat{t}_a^{(i,r)}\right) \tag{5.3.13}$$

$$\widetilde{b}^{(i,r)} \leftarrow (1-\rho)\widetilde{b}^{(i,r)} + \rho\left(b^{(i)} + s \cdot \hat{t}_b^{(i,r)}\right) \tag{5.3.14}$$

for $i = 1, 2, \ldots, N$ and $r = 1, 2, \ldots, r_{\max}$, where

$$\hat{t}_a^{(i,r)} \triangleq \frac{1}{S}\sum_{\hat{x}\in\mathcal{S}} \sum_{d\in d^{(i)}(x)} (d-1) \tag{5.3.15}$$

$$\hat{t}_b^{(i,r)} \triangleq \frac{1}{S}\sum_{\hat{x}\in\mathcal{S}} \sum_{d\in d^{(i)}(\hat{x})} r \tag{5.3.16}$$

$$\begin{aligned}(\hat{t}_r^{(i)})_r &\triangleq \mathbb{E}_{q(p|r)\hat{q}(x_{1:T})}\left[t_r(p, d^{(i)}(\hat{x}_{1:T}))\right] \qquad (5.3.17)\\ &= \left(\widetilde{a}^{(i,r)} + \hat{t}_a^{(i,r)} - 1\right)\mathbb{E}_{q(p|r)}\left[\ln(p^{(i,r)})\right] + \left(\widetilde{b}^{(i,r)} + \hat{t}_b^{(i,r)} - 1\right)\mathbb{E}_{q(p|r)}\left[\ln(1-p^{(i,r)})\right]\\ &\quad + \sum_{\hat{x}\in\mathcal{S}}\sum_{d\in d^{(i)}(\hat{x})} \ln\binom{d+r-2}{d-1}. \qquad (5.3.18)\end{aligned}$$

Similarly, we revise Eqs. (5.2.8)-(5.2.10) to compute the other expected sufficient statis-

**Algorithm 5.5** Negative Binomial HSMM SVI

Initialize global variational parameters $\widetilde{\eta}_\vartheta^{(i)}$, $\widetilde{\eta}_\theta^{(i)}$, $\widetilde{\alpha}^{(i)}$, and $\widetilde{\alpha}^{(0)}$
**for** $t = 1, 2, \ldots$ **do**
  Sample minibatch index $\hat{k}$ uniformly from $\{1, 2, \ldots, K\}$
  Using minibatch $\bar{y}^{(\hat{k})}$, generate state sequence samples
    according to Eqs. (5.3.10) and (5.3.11) and form $\hat{q}(x_{1:T})$
  Using $\hat{q}(x_{1:T})$, compute each $\hat{t}_{\text{dur}}^{(i)}$, $\hat{t}_y^{(i)}$, $\hat{t}_{\text{trans}}^{(i)}$, and $\hat{t}_{\text{init}}^{(i)}$
    with Eqs. (5.3.19)-(5.3.21) and (5.3.15)-(5.3.18)
  Update each $\widetilde{\eta}_\vartheta^{(i)}$, $\widetilde{\eta}_\theta^{(i)}$, $\widetilde{\alpha}^{(i)}$, and $\widetilde{\alpha}^{(0)}$
    with Eqs. (5.2.11)-(5.2.13) and (5.3.12)-(5.3.14)

tics using $\hat{q}(x_{1:T})$:

$$\hat{t}_y^{(i)} \triangleq \mathbb{E}_{\hat{q}(x_{1:T})} \sum_{t=1}^{T} \mathbb{I}[x_t = i] t_y^{(i)}(\bar{y}_t) \tag{5.3.19}$$

$$(\hat{t}_{\text{trans}}^{(i)})_j \triangleq \mathbb{E}_{\hat{q}(x_{1:T})} \sum_{t=1}^{T-1} \mathbb{I}[x_t = i, x_{t+1} = j] \tag{5.3.20}$$

$$(\hat{t}_{\text{init}})_i \triangleq \mathbb{E}_{\hat{q}(x_{1:T})} \mathbb{I}[x_1 = i] \tag{5.3.21}$$

We summarize the overall algorithm in 5.5.

## ■ 5.4 Extending to the HDP-HMM and HDP-HSMM

In this section we extend our methods to the Bayesian nonparametric versions of these models, the HDP-HMM and the HDP-HSMM. These updates essentially replace the transition updates in the previous algorithms.

Using the notation of Section 2.5 the generative model for the HDP-HMM with scalar concentration parameters $\alpha, \gamma > 0$ is

$$\beta \sim \text{GEM}(\gamma), \qquad \pi^{(i)} \sim \text{DP}(\alpha\beta), \qquad \theta^{(i)} \overset{\text{iid}}{\sim} p(\theta^{(i)}) \tag{5.4.1}$$

$$x_1 \sim \pi^{(0)}, \qquad x_{t+1} \sim \pi^{(x_t)}, \qquad y_t \sim p(y_t|\theta^{(x_t)}) \tag{5.4.2}$$

where $\beta \sim \text{GEM}(\gamma)$ denotes sampling from a stick breaking distribution defined by

$$v_j \overset{\text{iid}}{\sim} \text{Beta}(1, \gamma), \qquad \beta_k = \prod_{j<k} (1 - v_j) v_k \tag{5.4.3}$$

and $\pi^{(i)} \sim \mathrm{DP}(\alpha\beta)$ denotes sampling a Dirichlet process

$$w \sim \mathrm{GEM}(\alpha) \qquad z_k \overset{\text{iid}}{\sim} \beta \qquad \pi^{(i)} = \sum_{k=1}^{\infty} w_k \delta_{z_k}. \tag{5.4.4}$$

To perform mean field inference in HDP models, we approximate the posterior with a truncated variational distribution. While a common truncation is to limit the two stick-breaking distributions in the definition of the HDP [52], a more convenient truncation for our models is the "direct assignment" truncation, used in [70] for batch mean field with the HDP-HMM and in [17] in an SVI algorithm for LDA. The direct assignment truncation limits the support of $q(x_{1:T})$ to the finite set $\{1, 2, \ldots, M\}^T$ for a truncation parameter $M$, i.e. fixing $q(x_{1:T}) = 0$ when any $x_t > M$. Thus the other factors, namely $q(\pi)$, $q(\beta)$, and $q(\theta)$, only differ from their priors in their distribution over the first $M$ components. As opposed to standard truncation, this family of approximations is nested over $M$, enabling a search procedure over the truncation parameter as developed in [17]. A similar search procedure can be used with the HDP-HMM and HDP-HSMM algorithms developed here.

A disadvantage to the direct assignment truncation is that the update to $q(\beta)$ is not conjugate given the other factors as in Hoffman et al. [52]. Following Liang et al. [70], to simplify the update we use a point estimate by writing $q(\beta) = \delta_{\beta^*}(\beta)$. Since the main effect of $\beta$ is to enforce shared sparsity among the $\pi^{(i)}$, it is reasonable to expect that a point approximation for $q(\beta)$ will suffice.

The updates to the factors $q(\theta)$ and $q(x_{1:T})$ are identical to those derived in the previous sections. To derive the SVI update for $q(\pi)$, we write the relevant part of the untruncated model and truncated variational factors as

$$p((\pi^{(i)}_{1:M}, \pi^{(i)}_{\text{rest}})) = \mathrm{Dir}(\alpha \cdot (\beta_{1:M}, \beta_{\text{rest}})) \tag{5.4.5}$$

$$q((\pi^{(i)}_{1:M}, \pi^{(i)}_{\text{rest}})) = \mathrm{Dir}(\widetilde{\alpha}^{(i)}) \tag{5.4.6}$$

where $i = 1, 2, \ldots, M$ and where $\pi^{(i)}_{\text{rest}} \triangleq 1 - \sum_{k=1}^{M} \pi^{(i)}_k$ and $\beta_{\text{rest}} \triangleq 1 - \sum_{k=1}^{M} \beta_k$. Therefore the updates to $q(\pi^{(i)})$ are identical to those in (5.2.12) except the number of variational parameters is $M + 1$ and the prior hyperparameters are replaced with $\alpha \cdot (\beta_{1:M}, \beta_{\text{rest}})$.

To derive a gradient of the variational objective with respect to $\beta^*$, we write

$$\nabla_{\beta^*}\mathcal{L} = \nabla_{\beta^*}\left\{\mathbb{E}_{q(\pi)}\left[\ln\frac{p(\beta,\pi)}{q(\beta)q(\pi)}\right]\right\} \tag{5.4.7}$$

$$= \nabla_{\beta^*}\left\{\ln p(\beta^*) + \sum_{i=1}^{M}\mathbb{E}_{q(\pi^{(i)})}\ln p(\pi^{(i)}|\beta^*)\right\} \tag{5.4.8}$$

where $\ln p(\beta^*) = \ln p_v(v(\beta^*)) + \ln\det\left.\frac{\partial v}{\partial\beta}\right|_{\beta^*}$, $\ln p_v(v) = (\gamma-1)\sum_j \ln(1-v_j)$, and $v_i(\beta) = \frac{\beta_i}{1-\sum_{j<i}\beta_j}$. The Jacobian $\frac{\partial v}{\partial\beta}$ is lower-triangular, and is given by

$$\left(\frac{\partial v}{\partial\beta}\right)_{ij} = \begin{cases} 0 & i<j \\ \frac{1}{1-\sum_{k<i}\beta_k} & i=j \\ \frac{-\beta_i}{(1-\sum_{k<i}\beta_k)^2} & i>j \end{cases} \tag{5.4.9}$$

and so taking partial derivatives we have

$$\frac{\partial}{\partial\beta_k^*}\ln p(\beta^*) = 2\sum_{i\geq k}\ln\frac{1}{1-\sum_{j<i}\beta_j^*} - (\gamma-1)\sum_{i\geq k}\ln\frac{1}{1-\sum_{j\leq i}\beta_j^*} \tag{5.4.10}$$

$$\frac{\partial}{\partial\beta_k^*}\mathbb{E}_{q(\pi)}[\ln p(\pi^{(i)}|\beta^*)] = \gamma\psi(\widetilde{\alpha}_k^{(i)}) - \gamma\psi(\widetilde{\alpha}_{M+1}^{(i)}) + \gamma\psi(\gamma\textstyle\sum_{j=1}^{M+1}\beta_j^*) - \gamma\psi(\beta_k^*). \tag{5.4.11}$$

We use this gradient expression to take a truncated gradient step on $\beta^*$ during each SVI update, where we use a backtracking line search[1] to ensure the updated value satisfies the constraint $\beta^* \geq 0$.

The updates for $q(\pi)$ and $q(\beta)$ in the HDP-HSMM differ only in that the variational lower bound expression changes slightly because the support of each $q(\pi^{(i)})$ is restricted to the off-diagonal (and renormalized). We can adapt $q(\pi^{(i)})$ by simply dropping the $i$th component from the representation and writing

$$q((\pi_{1:M\setminus i}^{(i)}, \pi_{\text{rest}}^{(i)})) = \text{Dir}(\widetilde{\alpha}_{\setminus i}^{(i)}), \tag{5.4.12}$$

and we change the second term in the gradient for $\beta^*$ to

$$\frac{\partial}{\partial\beta_k^*}\mathbb{E}_{q(\pi)}[\ln p(\pi^{(i)}|\beta^*)] = \begin{cases} \gamma\psi(\widetilde{\alpha}_k^{(i)}) - \gamma\psi(\widetilde{\alpha}_{M+1}^{(i)}) + \gamma\psi(\gamma\sum_{j\neq i}\beta_j^*) - \gamma\psi(\beta_k^*) & k\neq i \\ 0 & k=i \end{cases}. \tag{5.4.13}$$

Using these gradient expressions for $\beta^*$ and a suitable gradient-based optimization procedure we can also perform batch mean field updates for the HDP-HSMM.

[1] In a backtracking line search, for some fixed parameter $\kappa \in (0,1)$, given an initial point $x$ and an increment $\Delta$, while $x+\Delta$ is infeasible we set $\Delta \leftarrow \kappa\Delta$.

## 5.5 Experiments

We conclude this chapter with a numerical study to validate the proposed algorithms.

As a performance metric, we approximate a variational posterior predictive density on held-out data; that is, for the HMM models we estimate

$$p(\bar{y}_{\text{test}}|\bar{y}_{\text{train}}) = \int\int p(\bar{y}_{\text{test}}|\pi,\theta)p(\pi,\theta|\bar{y}_{\text{train}})\mathrm{d}\pi\mathrm{d}\theta \tag{5.5.1}$$

$$\approx \mathbb{E}_{q(\pi)q(\theta)}p(\bar{y}_{\text{test}}|\pi,\theta) \tag{5.5.2}$$

by sampling models from the fit variational distribution. Similarly, for HSMM models we estimate $p(\bar{y}_{\text{test}}|\bar{y}_{\text{train}}) \approx \mathbb{E}_{q(\pi)q(\theta)q(\vartheta)}p(\bar{y}_{\text{test}}|\pi,\theta,\vartheta)$. In each experiment, we chose $\rho^{(t)} = (t+\tau)^{-\kappa}$ with $\tau = 0$ and $\kappa = 0.6$. Gaussian emission parameters were generated from Normal-Inverse-Wishart (NIW) distributions with $\mu_0 = 0$, $\Sigma_0 = I$, $\kappa_0 = 0.1$, and $\nu_0 = 7$. For the HDP models, we set the truncation parameters to be twice the true number of modes. Every SVI algorithm examined uses only a single pass through the training data.

First, we compare the performance of SVI and batch mean field algorithms for the HDP-HMM on synthetic data with fully conjugate priors. We sampled a 10-state HMM with 2-dimensional Gaussian emissions and generated a dataset of 250 sequences of length 4000 for a total of $10^6$ frames. We chose a random subset of 95% of the generated sequences to be training sequences and held out 5% as test sequences. We repeated the fitting procedures on the training set 5 times with initializations drawn from the prior, and we report the average performance with standard deviation error bars. In Figure 5.2, the SVI procedure (in blue) produces fits that are on par with those from the batch algorithm (in green) but orders of magnitude faster. In particular, note that the SVI algorithm consistently converges to a local optimum of the mean field objective in a single pass through the training set, requiring roughly the amount of time needed for a single iteration of the batch mean field algorithm. This relative speedup grows linearly with the size of the dataset, making the SVI algorithm especially useful when the batch algorithm is infeasible.

Similarly, we compare the SVI and batch mean field algorithms for the HDP-HSMM. We sampled a 6-state HSMM with 2-dimensional Gaussian emissions and negative binomial durations, where each of the negative binomial parameters were sampled as $p \sim \text{Beta}(1,1)$ and $r \sim \text{Uniform}(\{1,2,\ldots,10\})$. From the model we generated a dataset of 50 sequences of length 2000 and generated an additional test set of 5 sequences with the same length. Figure 5.3 shows again that the SVI procedure (in blue) fits the data orders of magnitude faster than the batch update (in green), and again it requires only a single pass through the training set.

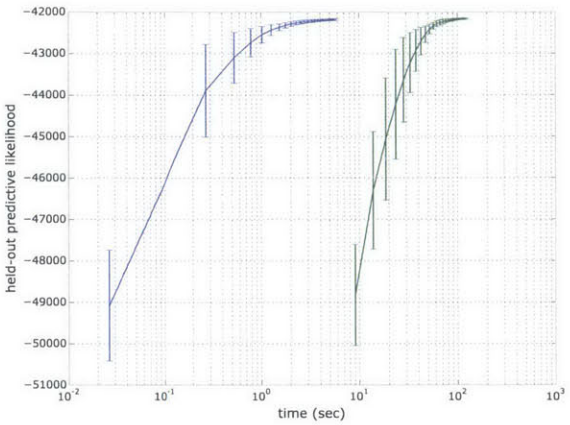



**Figure 5.2:** A comparison of the HMM SVI algorithm with batch mean field. Algoritm 5.3 is shown in blue and the batch mean field algorithm is shown in green.

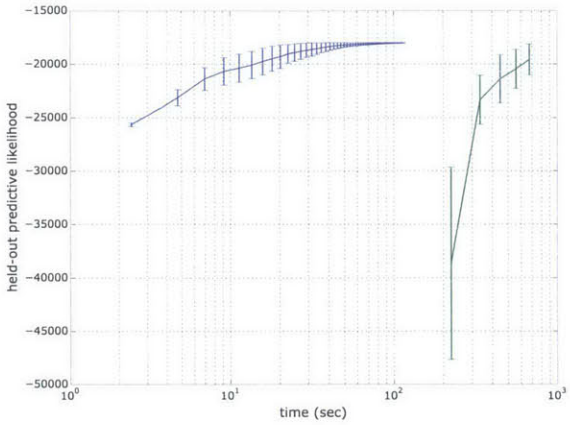



**Figure 5.3:** A comparison of the HSMM SVI algorithm with batch mean field. Algorithm 5.4 is shown in blue and the batch mean field algorithm is shown in green.

Finally, we compare the performance of the exact SVI update for the HSMM with that of the approximate update proposed in Section 5.3. We sampled a 6-state HSMM with 2-dimensional Gaussian emissions and Poisson durations, where each of the Poisson duration parameters is sampled as $\lambda \sim \text{Gamma}(40, 2)$. From the model we generated a dataset of 50 sequences of length 3000 and generated an additional test set of 5 sequences with the same length. We fit the data with negative binomial HDP-HSMMs where the priors on the negative binomial parameters were again $p \sim \text{Beta}(1, 1)$ and $r \sim \text{Uniform}(\{1, 2, \ldots, 10\})$. We set the number of state sequence samples generated in

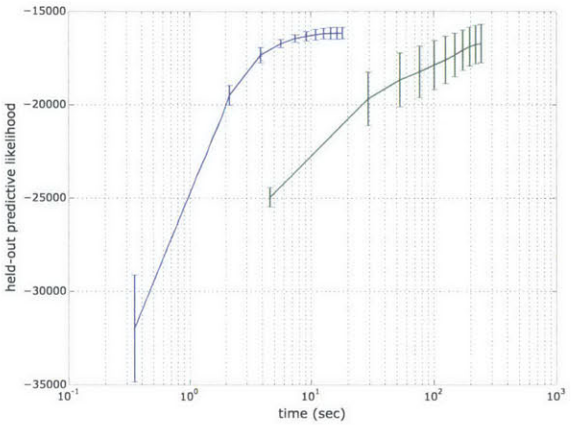



**Figure 5.4:** A comparison of HSMM SVI algorithms. The approximate update scheme of Algorithm 5.5 is shown in blue and the exact update scheme of Algorithm 5.4 is shown in green.

the sampling-based approximate update to $S = 10$. Figure 5.4 shows that the sampling-based updates (in blue) are effective and that they provide a significant speedup over the exact SVI update (in green). Note that, since the figure compares two SVI algorithms, both algorithms scale identically with the size of the dataset. However, the time required for the exact update scales quadratically with the minibatch sequence length $T$, while the sampling-based update scales only linearly with $T$. Therefore this approximate SVI update is most useful when minibatch sequences are long enough so that the exact update is infeasible.

Chapter 6

# Scalable Inference in Models with Multiple Timescales

## 6.1 Introduction

In many settings we may wish to learn dynamics at multiple timescales. For example, in the context of speech analysis, we may wish to model both the dynamics within individual phonemes as well as the dynamics across phonemes [68, 18]. In the context of modeling behavior, motion [51], or handwriting [67], it is natural to decompose movements into steps, while still modeling the statistics of the sequence of movements. Each of these modeling tasks involves dynamics at multiple timescales, and therefore it is natural to consider dynamical models that can capture such dynamics while maintaining tractable inference.

In this chapter, we develop a Bayesian nonparametric model and associated inference algorithms applicable to unsupervised learning of such dynamics. We combine and build on ideas developed in previous chapters. In particular, we extend the HDP-HSMM developed in Chapter 3 to include Markovian dynamics within each of its segments. The explicit duration modeling provided by the HDP-HSMM allows us to set duration priors that can disambiguate short-timescale dynamics from long-timescale dynamics and is important for identifiability in the unsupervised setting. Using ideas from Chapters 3 and 4, we develop efficient Gibbs sampling algorithms for our proposed model. Finally, extending ideas from Chapter 5, we also develop a structured Stochastic Variational Inference (SVI) algorithm, which allows inference to scale to large datasets. Developing scalable inference with efficient updates is particularly relevant when fitting rich models, since more data are often required to fit more complex models effectively.

The main contributions of this chapter are algorithmic, particularly in incorporating the algorithmic techniques developed in previous chapters. While the model we propose is new, as we discuss in Section 6.2 many similar models have been explored in the literature. The key advantage to our model is its amenability to the efficient inference algorithms we develop. Our model also benefits from explicit duration modeling and a

Bayesian nonparametric definition, which enable both explicit control over important priors and flexible learning.

In Section 6.2 we highlight some key related work. In Section 6.4 we develop several Gibbs sampling algorithms for the model, including a collapsed direct assignment sampler, a weak limit sampler, and a more efficient weak limit sampler when durations are modeled with negative binomial distributions. In Section 6.5 we develop mean field and SVI updates. Finally, in Section 6.6, we demonstrate our algorithms with an application to unsupervised phoneme discovery.

This chapter synthesizes and extends results from previous chapters and so we rely heavily on their notation and content.

## ■ 6.2 Related work

The model we define in this chapter is most closely related to generalizations of HMMs and HSMMs known as segment models, which can model sub-dynamics within an HMM or HSMM state. Segment models have a long history in the HMM literature; see Murphy [78] and Murphy [80, Section 17.6] and the references therein. Such models have had considerable success in modeling multiscale dynamics, particular in modeling speech dynamics at the level of words, phones, and sub-phones [80, p. 624]. Such models have typically been explored in non-Bayesian settings. Our model can be viewed as a Bayesian nonparametric segment model, where the Bayesian approach gives us explicit control over duration priors and modeling of uncertainty, and the nonparametric definition provides for flexible learning of model complexity.

A related class of models is the class of Hierarchical HMMs (HHMMs) [28] [80, Section 17.6.2], which have also been studied extensively in non-Bayesian settings. A Bayesian nonparametric HHMM, the infinite HHMM (iHHMM), has been developed and applied successfully to some small example datasets [51]. The model represents an infinite number of dynamical timescales and is extremely flexible. However, it does not provide explicit duration modeling and so it is not easy to use priors to control timescales in the learned dynamics. Furthermore, its structure is not particularly amenable to scalable inference, and in its Gibbs sampling algorithm the hidden states at each level must be sampled conditioned on the hidden states at all the other levels. The model we propose has only two timescales and so is less flexible than the iHHMM, but it allows explicit prior control over duration distributions. In addition, the algorithms we develop exploit more powerful message passing, and the SVI algorithm developed in Section 6.5 allows inference in our model to scale to large datasets.

Finally, the model that is most similar to ours is that of Lee and Glass [68], which develops a Bayesian nonparametric model for unsupervised phoneme discovery. Again, a key difference is again that our model provides explicit duration modeling and al-

lows much more scalable algorithms. In addition, we allow for the substate dynamics themselves to be modeled nonparametrically, while the model of Lee and Glass [68] focuses on modeling each phoneme with fixed-size finite HMMs. While our model can also use fixed-size finite HMMs for short-timescale dynamics, we focus on the fully nonparametric specification.

## 6.3 Model specification

In this section, we define our generative model, composing both the HDP-HSMM and HDP-HMM generative processes described in Chapter 3, particularly Sections 3.2.3 and 3.3.2. Recall that we write the prior measure on duration parameters as $G$ and the corresponding duration likelihood as $p(d|\vartheta^{(i)})$, and let $\alpha, \gamma > 0$ be concentration parameters. According to the HDP-HSMM generative process, we generate the super-state sequence $(z_s)$, the duration sequence $(d_s)$, and the label sequence $(x_t)$ as

$$\beta \sim \text{GEM}(\gamma) \tag{6.3.1}$$

$$\pi^{(i)} \overset{\text{iid}}{\sim} \text{DP}(\alpha, \beta) \qquad \vartheta^{(i)} \overset{\text{iid}}{\sim} G \qquad i = 1, 2, \ldots \tag{6.3.2}$$

$$z_s \sim \bar{\pi}^{(z_{s-1})} \qquad d_s \sim p(d|\vartheta^{(z_s)}) \qquad s = 1, 2, \ldots \tag{6.3.3}$$

$$x_{t(s):t(s+1)-1} = z_s \qquad t(s) \triangleq \begin{cases} t(s-1) + d_{s-1} & s > 1 \\ 1 & s = 1 \end{cases} \qquad t = 1, 2, \ldots, T, \tag{6.3.4}$$

where, as in Section 3.3.2, $\bar{\pi}^{(i)} \triangleq \frac{\pi_j^{(i)}}{1-\pi_i^{(i)}}(1 - \delta_{ij})$. While the HDP-HSMM generates the observation sequence $(y_t)$ within a segment as conditionally independent draws from an observation distribution, here we instead generate observations for each segment according to an HDP-HMM. That is, for each HDP-HSMM state $i = 1, 2, \ldots$ we have an HDP-HMM with parameters $\{\beta^{(i)}, \pi^{(i,j)}, \theta^{(i,j)}\}_{j=1}^{\infty}$ generated according to

$$\beta^{(i)} \sim \text{GEM}(\gamma^{(i)}) \tag{6.3.5}$$

$$\pi^{(i,j)} \overset{\text{iid}}{\sim} \text{DP}(\alpha^{(i)}, \beta^{(i)}) \qquad \theta^{(i,j)} \overset{\text{iid}}{\sim} H \qquad j = 1, 2, \ldots \tag{6.3.6}$$

where $H$ is the prior measure over observation parameters $\theta^{(i,j)}$, $\alpha^{(i)}$ and $\gamma^{(i)}$ are concentration parameters, and each $\pi^{(i,j)}$ is a transition distribution out of the corresponding HDP-HMM's $j$th state. Then for a segment $s$ in with HDP-HSMM super-state $z_s$ and duration $d_s$, we generate observations from the corresponding HDP-HMM via

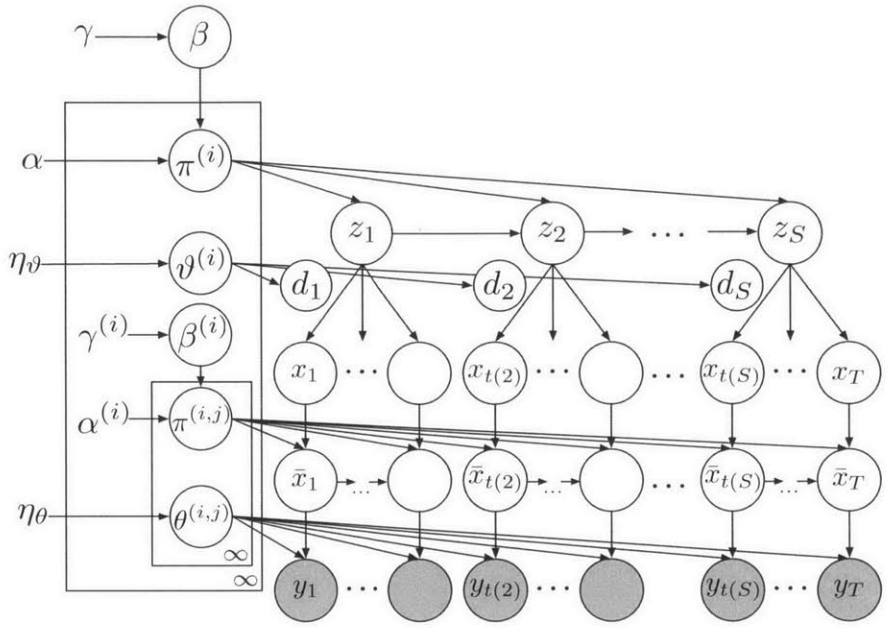



**Figure 6.1:** A graphical model for the HDP-HSMM with sub-HDP-HMM observations. Note that it is not formally a graphical model because the number of nodes is random due to the random durations.

$$\bar{x}_t \sim \pi^{(z_s,\bar{x}_{t-1})} \qquad y_t \sim p(y|\theta^{(z_s,\bar{x}_t)}) \qquad t = t(s), t(s)+1, \ldots, t(s+1)-1, \tag{6.3.7}$$

where $\theta^{(z_s,\bar{x}_t)}$ is the observation parameter from the corresponding HDP-HMM's $j$th state, and $p(y|\theta^{(z_s,\bar{x}_t)})$ is the corresponding observation likelihood. We call $(\bar{x}_t)_{t=1}^T$ the *substate sequence*, and emphasize that it is distinct from the HDP-HSMM's super-state sequence $(z_s)$ and label sequence $(x_t)$. See Figure 6.1 for a graphical model.

This model definition combines the explicit duration modeling and nonparametric flexibility of the HDP-HSMM of Chapter 3 with HDP-HMM dynamics within each HSMM segment. The HDP-HSMM states can model longer-timescale dynamics, such as the dynamics between phonemes, while the HDP-HMM states can model shorter-timescale dynamics, such as the structure within an individual phoneme. As we show in the following sections, this model definition is also amenable to efficient inference.

While we have defined this model using Bayesian nonparametric priors for both layers of dynamics, it is straightforward to adapt the definition so that one or both of the layers is finite. For example, it may be desirable to use finite structured sub-HMM models to exploit some domain knowledge [68]. Alternatively, it is also possible to make the coarser-scale dynamical process a finite HSMM while allowing the substate

dynamics to be generated from an HDP-HMM, thus enabling model selection via a semiparametric approach [38].

## ■ 6.4 Gibbs sampling

In this section, we develop several Gibbs sampling algorithms for the model defined in Section 6.3. First, we develop a collapsed direct assignment sampler. This sampler avoids approximating the posterior with a finite distribution but, as with the direct assignment sampler developed in Chapter 3 and simulated in Figure 3.11(b), its mixing rate is far too slow to be practically useful. We include it for completeness and theoretical interest.

Second, we develop a sampler based on the weak limit approximation. Analogous to the weak limit sampler developed in Chapter 3, this sampler can use message passing to perform block sampling and therefore achieve much greater mixing.

Finally, we build on the results of Chapter 4 to develop a much faster weak limit Gibbs sampler for negative binomial duration distributions. The message passing complexity is greatly reduced, and in particular is only linear in the sequence length $T$.

### ■ 6.4.1 Collapsed Gibbs sampler

To develop a collapsed Gibbs sampler, we extend the HDP-HSMM direct assignment Gibbs algorithm developed in Section 3.4.3. Essentially, we combine the HDP-HSMM direct assignment sampler and the HDP-HMM direct assignment sampler.

The algorithm state for our direct assignment sampler consists of a finite prefix of the HDP-HSMM $\beta$ parameter and finite prefixes of each of the sub-HDP-HMM $\beta^{(i)}$ parameters. It also includes both the HDP-HSMM label sequence $(x_t)$ and the substate sequence $(\bar{x}_t)$. That is, we write the sampler state as $(\beta_{1:N}, \beta^{(i)}_{1:N^{(i)}}, x_{1:T}, \bar{x}_{1:T})$, where we use $N$ to represent the number of used HDP-HSMM states and $N^{(i)}$ to represent the number of used states in the $i$th HDP-HMM. The other parameters are integrated out analytically, including the HDP-HSMM transition parameters $\{\pi^{(i)}\}$, the sub-HDP-HMM transition parameters $\{\pi^{(i,j)}\}$, the duration parameters $\{\vartheta^{(i)}\}$, and the observation parameters $\{\theta^{(i,j)}\}$.

The algorithm proceeds by jointly resampling each pair $(x_t, \bar{x}_t)$ for $t = 1, 2, \ldots, T$ and by resampling the $\beta_{1:N}$ and $\beta^{(i)}_{1:N^{(i)}}$.

**Resampling the label and substate sequence.** To resample each $(x_t, \bar{x}_t)$ conditioned on all the other variables and parameters, we extend the HDP-HSMM sampling step of Section 3.4.3 That is, we resample $(x_t, \bar{x}_t)$ by considering all possible assignments $(k, k')$ for $k = 1, 2, \ldots, N+1$ and $k' = 1, 2, \ldots, N^{(i)} + 1$ and evaluating up to proportionality the conditional probability

$$p((x_t, \bar{x}_t) = (k, k')|(x_{\setminus t}), (\bar{x}_{\setminus t}), \beta, \{\beta^{(i)}\}), \tag{6.4.1}$$

where we suppress notation for conditioning on all the hyperparameters. Recall that as we vary the assignment of the HDP-HSMM label $x_t$ we must consider the possible merges into adjacent label segments, and as a result there are between 1 and 3 terms in the expression for (6.4.1), each of the form

$$p((x_t, \bar{x}_t) = (k, k')|(x_{\setminus t}, \bar{x}_{\setminus t})) \propto \underbrace{\frac{\alpha\beta_k + n_{x_{\text{prev}},k}}{\alpha(1-\beta_{x_{\text{prev}}}) + n_{x_{\text{prev}},\cdot}}}_{\text{left-transition}} \cdot \underbrace{\frac{\alpha\beta_{x_{\text{next}}} + n_{k,x_{\text{next}}}}{\alpha(1-\beta_k) + n_{k,\cdot}}}_{\text{right-transition}}$$
$$\cdot \underbrace{f_{\text{dur}}(t_2 - t_1 + 1)}_{\text{duration}} \cdot \underbrace{f_{\text{obs}}(y_{t_1:t_2}|k)}_{\text{observation}}, \tag{6.4.2}$$

where we have used $t_1$ and $t_2$ to denote the first and last indices of the segment, respectively, and $(x_{\setminus t})$ and $(\bar{x}_{\setminus t})$ to denote the label sequence and substate sequence assignments excluding the $t$th index, respectively. See Section 3.4.3 for details. In the case of the HDP-HSMM of Chapter 3, the term $f_{\text{obs}}(y_{t_1:t_2}|k)$ is computed from the independent observation model. For sub-HDP-HMM observations, we simply replace this term with the appropriate score for HDP-HMM observations, incorporating not only the data $y_{t_1:t_2}$ but also the substate assignment $\bar{x}_t = k'$ and the substate sequence $(\bar{x}_{\setminus t})$.

Using the formula for the probability of an assignment sequence under an HDP model [43, 106], we can write the $f_{\text{obs}}(y_{t_1:t_2}|k, k')$ term for sub-HDP-HMM observations. Let $n_{ij}$ be the number of transitions from substate $i$ to substate $j$ in the $k$th sub-HMM, and let $n_{i\cdot} = \sum_{j=1}^{N^{(k)}} n_{ij}$. In addition, for first and last segment indices $t_1$ and $t_2$, let

$$\bar{n}_{ij} = n_{ij} - \#\{t : \bar{x}_t = i, \bar{x}_{t+1} = j, t_1 \leq t < t_2, x_t = k\} \tag{6.4.3}$$

be the number of transitions from substate $i$ to substate $j$ in the $k$th sub-HMM excluding those in the segment from $t_1$ to $t_2$, and let $\bar{n}_{i\cdot} = \sum_{j=1}^{N^{(k)}} \bar{n}_{ij}$. Then we can write

$$f_{\text{obs}}(y_{t_1:t_2}|k,k') = \left(\prod_{i=1}^{N^{(k)}} \frac{\Gamma(\alpha+\bar{n}_{i\cdot})}{\Gamma(\alpha+n_{i\cdot})} \prod_{j=1}^{N^{(k)}} \prod_{\ell=\bar{n}_{ij}}^{n_{ij}-1} (\alpha\beta_j^{(k)}+\ell)\right) \cdot \left(\int \prod_{t=t_1}^{t_2} p(y_t|\theta^{(k,k')}) p(\theta^{(k,k')}|\{y_{t'} : \bar{x}_{t'}=k', x_{t'}=k\}, \eta_\theta)\, d\theta^{(k,k')}\right) \tag{6.4.4}$$

where $\eta_\theta$ is the corresponding observation hyperparameter. By substituting (6.4.4) for $f_{\text{obs}}$ in (6.4.2), we can proceed with the HDP-HSMM sampling procedure of Section 3.4.3.

**Resampling $\beta_{1:N}$ and $\beta^{(i)}_{1:N^{(i)}}$.** To resample $\beta_{1:N}$ conditioned on the HDP-HSMM label sequence $(x_t)$, we use the same auxiliary variable method developed in Section 3.4.2. To resample each $\beta^{(i)}_{1:N^{(i)}}$ for each sub-HDP-HMM, $i = 1, 2, \ldots, N$, we use the standard HDP-HMM direct assignment update [106].

## ■ 6.4.2 Weak limit sampler

We can develop a more efficient sampling algorithm by using a weak limit approximation and exploiting dynamic programming. In particular, we develop a weak limit sampler that block resamples the label sequence and substate sequence jointly. We build on the weak limit sampler developed in Section 3.4.2. We write the weak limit truncation level of the HDP-HSMM as $N$ and the weak limit truncation level of the $i$th HDP-HMM as $N^{(i)}$.

Recall that the label sequence $(x_t)$ can be resampled by first passing HSMM messages backward and then sampling forward, as in Section 3.4.1, particularly equations (3.4.3) and (3.4.6). From Section 3.2.2, the HSMM messages $(B, B^*)$ are defined by

$$B_{t,i} = \sum_{j=1}^{N} B^*_{t,j} p(x_{t+1}=j|x_t=i, x_{t+1}\neq x_t) = \sum_{j=1}^{N} B^*_{t,j} A_{ij}, \tag{6.4.5}$$

$$B^*_{t,i} = \sum_{d=1}^{T-t} B_{t+d,i} \underbrace{p(d_{s(t)}=d|z_{s(t+1)}=i)}_{\text{duration prior term}} \cdot \underbrace{p(y_{t+1:t+d}|z_{s(t+1)}=i, d_{s(t+1)}=d)}_{\text{likelihood term}} \tag{6.4.6}$$

$$= \sum_{d=1}^{T-t-1} B_{t+d,i} D_{d,i} p(y_{t+1:t+d}|z_{s(t+1)}=i, d_{s(t+1)}=d) \tag{6.4.7}$$

$$B_{T,i} \triangleq 1, \tag{6.4.8}$$

where

$$D_{d,i} = p(d|\vartheta^{(i)}) \qquad A_{ij} = p(x_{t+1} = j|x_t = i, x_{t+1} \neq x_t) \tag{6.4.9}$$

and where $s(t)$ denotes the segment index for time index $t$. In Chapter 3, the likelihood term $p(y_{t+1:t+d}|z_{s(t+1)} = i, d_{s(t+1)} = d)$ is computed as a product of independent likelihoods, while here we must compute it according to the HMM observation model. If we can compute these segment likelihood terms efficiently, we can use them in Eqs. (6.4.5)-(6.4.8) to compute the backward messages over the HSMM and sample the HSMM label sequence as in Section 3.4.1. Given the HSMM label sequence, we can then sample the substate sequence using the HMM state sequence sampling algorithm given in Section 2.4.1.

Therefore it remains only to compute the segment likelihood terms efficiently. We can exploit the Markov structure in the substate sequence to write these likelihoods in terms of another set of messages. For each $i = 1, 2, \ldots, N$, we define the sub-HMM backward messages for each $t = 1, 2, \ldots, T$ as

$$B_{t',j}^{(i,t)} = \sum_{k=1}^{N^{(i)}} A_{jk}^{(i)} L_{t'+1,k}^{(i)} B_{t'+1,k}^{(i,t)} \qquad t' = 1, 2, \ldots, t \qquad B_{t,j}^{(i,t)} = 1, \tag{6.4.10}$$

where $L_{t,k}^{(i)} = p(y_t|\theta^{(i,k)})$ is the observation likelihood for the $j$th substate of the $i$th HMM and $A_{jk}^{(i)} = p(\bar{x}_{t+1} = k|\bar{x}_t = j, x_t = i)$ is the probability of transitioning from the $j$th to the $k$th substate in the $i$th HMM. Similarly, we define the sub-HMM forward messages for each $i = 1, 2, \ldots, N$ and $t = 1, 2, \ldots, T$ as

$$F_{t',j}^{(i,t)} = \sum_{k=1}^{N^{(i)}} A_{kj} L_{t',j} F_{t'-1,k}^{(i,t)} \qquad t = t+1, t+2, \ldots, T \qquad F_{t,k}^{(i,t)} = \pi_k^{(i,0)} \tag{6.4.11}$$

where $\pi^{(i,0)}$ is the initial state distribution for the $i$th sub-HMM. For any fixed time index $t$ and superstate index $i$, we can compute these messages in time $\mathcal{O}(TN^{(i)^2})$ time, and therefore we can compute all such sub-HMM messages in time $\mathcal{O}(NT^2N^{(i)^2})$ time. Finally, we can use these messages to compute every segment likelihood term via

$$p(y_{t:t+d-1}|z_{s(t)} = i, d_{s(t)} = d) = \sum_{j=1}^{N^{(i)}} F_{t',j}^{(i,t)} B_{t',j}^{(i,t+d)}, \tag{6.4.12}$$

for any $t' = t, t+1, \ldots, t+d-1$. To compute only the backward HSMM messages, as required for the block sampling procedure, it suffices to compute only the sub-HMM

forward messages.

Composing these expressions, we can write the overall HSMM messages as

$$B^*_{t,i} = \sum_{d=1}^{T-1-t} B_{t+d,i} D_{d,i} \left( \sum_{\ell=1}^{N^{(i)}} F^{(i,t)}_{t+d,\ell} \right) \qquad B_{t,i} = \sum_{j=1}^{N} A_{ij} B^*_{t,j}. \tag{6.4.13}$$

Writing $N_{\text{sub}} = \max_i N^{(i)}$, these messages require $\mathcal{O}(T^2 N N_{\text{sub}}^2 + TN^2)$ time to compute. With these messages, we can block resample the HSMM label sequence and substate sequence.

The sampling updates to the other model parameters are identical to those described in Sections 3.4.1 and 2.4.1.

## ■ 6.4.3 Exploiting negative binomial durations

While the weak limit sampling procedure developed in Section 6.4.2 is general, it can be computationally expensive for long observation sequences. In this section we apply and extend the ideas developed in Chapter 4 to write an update for negative binomial duration models for which the computational complexity scales only linearly in $T$ and is generally much more efficient. As in Chapter 4, this algorithm generalizes immediately to models in which the duration distributions are mixtures of negative binomial distributions.

Recall from Chapter 4 that with negative binomial durations we can compute the HSMM messages with more efficient recursions because the duration can be represented as an augmentation with a small number of Markov states. In particular, to represent an HSMM with negative binomial parameters $(r^{(i)}, p^{(i)})$ for $i = 1, 2, \ldots, N$, we constructed an equivalent HMM on $\sum_{i=1}^N r^{(i)}$ states. We can similarly embed an HSMM with HMM emissions and negative binomial durations in a stationary HMM on $\sum_{i=1}^N r^{(i)} N^{(i)}$ states. Using the notation of Chapter 4, we choose

$$\bar{A}^{(i)} = \hat{A}^{(i)} \otimes A^{(i)}, \qquad \bar{b}^{(i)} = \hat{b}^{(i)} \otimes \mathbb{1}^{(i)}, \qquad \bar{c}^{(i)} = \hat{c}^{(i)} \otimes \pi^{(i)}_{\text{sub}}, \tag{6.4.14}$$

where $X \otimes Y$ denotes the Kronecker product of matrices $X$ and $Y$, $\mathbb{1}^{(i)}$ denotes the all-ones vector of size $r^{(i)}$, and $(\hat{A}^{(i)}, \hat{b}^{(i)}, \hat{c}^{(i)})$ denotes the HMM embedding parameters for negative binomial durations given in Eqs. (4.4.7)-(4.4.8). Note that we can index into the matrix $\bar{A}$ using the tuple $(i, j, k)$, where $i = 1, 2, \ldots, N$ indexes the HSMM state, $j = 1, 2, \ldots, r^{(i)}$, indexes the duration pseudostate, and $k = 1, 2, \ldots, N^{(i)}$ indexes the sub-HMM substate. By comparing this construction to that of the embedding developed in Section 4.4, it is clear that it encodes the same dynamics on the HSMM label sequence: if we simply sum over the sub-HMM substates, we recover the same

embedding as that of Section 4.4. That is, if we use $\hat{A}$ to denote the transition matrix of the HMM embedding of Section 4.4, then

$$\hat{A}_{(i,j),(i',j')} = \sum_{k=1}^{N^{(i)}} \sum_{k'=1}^{N^{(i')}} \bar{A}_{(i,j,k),(i',j',k')}. \tag{6.4.15}$$

Furthermore, the sub-HMM substate dynamics are faithfully represented: the last block row of $\bar{A}^{(i,j)}$ ensures that the first substate of a segment is sampled according to $\pi^{(j)}$, and the substate transition probabilities are those of $A^{(i)}$ until the superstate changes.

Note that due to the structure of the matrix $\bar{A}$, the matrix-vector multiplications required to perform message passing are especially efficient to compute. In particular, using the identity

$$(X \otimes Y)\mathrm{vec}(Z) = \mathrm{vec}(XZY^{\mathsf{T}}) \tag{6.4.16}$$

for any matrices $X$, $Y$, and $Z$ of appropriate dimensions, we can compute the block diagonal part of a matrix-vector product in $\mathcal{O}(N(R + N_{\text{sub}}^2))$, where $R = \max_i r^{(i)}$. Furthermore, using the structure in each $\bar{A}^{(i,j)}$, we can compute the off-block-diagonal part of a matrix-vector product in $\mathcal{O}(N^2 + NN_{\text{sub}})$. Therefore, using the methods developed in Chapter 4, we can use the embedding to compute the HSMM messages in only $\mathcal{O}(TN(R + N_{\text{sub}}^2) + TN^2)$ time, avoiding the quadratic dependence on $T$. Finally, note that, using the methods developed in Chapter 4, this HSMM messages computation requires only $\mathcal{O}(TN + NRN_{\text{sub}})$ memory, a significant savings compared to the $\mathcal{O}(TNRN_{\text{sub}})$ memory required to compute the HMM messages in the full HMM embedding.

Given the HSMM messages, we can perform the block sampling update to the label sequence and substate sequence described in Section 6.4.2 much more efficiently.

## ■ 6.5 Mean field and SVI

In this section, we derive the key updates necessary to perform mean field or SVI inference in the model. This section relies heavily on the notation used in Chapter 5, and extends its results to the model developed in this chapter.

Following the notation of Chapter 5, we write our variational family as

$$q(\beta)q(\pi^{(0)})\prod_{i=1}^{N} q(\pi^{(i)})q(\vartheta^{(i)})\left(q(\beta^{(i)})q(\pi^{(i,0)})\prod_{j=1}^{N^{(i)}} q(\pi^{(i,j)})q(\theta^{(i,j)})\right) q(x_{1:T}, \bar{x}_{1:T}) \tag{6.5.1}$$

where $N$ is the truncation parameter for the HDP-HSMM and each $N^{(i)}$ is the truncation parameter for the $i$th sub-HDP-HMM. The variational factors are defined analo-

gously to those used in Chapter 5:

$$q(\theta^{(i,j)}) \propto \exp\left\{\langle \widetilde{\eta}_\theta^{(i,j)}, t_\theta^{(i,j)}(\theta^{(i,j)})\rangle\right\} \qquad q(\pi^{(i,j)}) = \mathrm{Dir}(\widetilde{\alpha}^{(i,j)}) \tag{6.5.2}$$

$$q(\pi^{(i)}) = \mathrm{Dir}(\widetilde{\alpha}^{(i)}) \qquad q(\vartheta^{(i)}) \propto \exp\left\{\langle \widetilde{\eta}_\vartheta^{(i)}, t_\vartheta^{(i)}(\vartheta^{(i)})\rangle\right\} \tag{6.5.3}$$

$$q(\beta) = \delta_{\beta^*}(\beta) \qquad q(\beta^{(i)}) = \delta_{\beta^{*(i)}}(\beta^{(i)}). \tag{6.5.4}$$

The corresponding prior densities on each term are

$$p(\theta^{(i,j)}) \propto \exp\left\{\langle \eta_\theta^{(i)}, t_\theta^{(i)}(\theta^{(i,j)})\rangle\right\} \qquad p(\pi^{(i,j)}|\beta_{1:N^{(i)}}) = \mathrm{Dir}(\alpha^{(i)}\beta_{1:N^{(i)}}) \tag{6.5.5}$$

$$p(\pi^{(i)}|\beta_{1:N}) = \mathrm{Dir}(\alpha\beta_{1:N}) \qquad p(\vartheta^{(i)}) \propto \exp\left\{\langle \eta_\vartheta^{(i)}, t_\vartheta^{(i)}(\vartheta^{(i)})\rangle\right\}. \tag{6.5.6}$$

We derive a mean field update to the variational factors over model parameters in two steps: first, we define structured mean field message-passing recursions analogous to those defined in Section 6.4.2; second, we show how to use the mean field messages to compute the expected statistics necessary for the parameter updates.

As in Section 6.4.2, it is useful to define sub-HMM messages for each $i = 1, 2, \ldots, N$ and each time index $t = 1, 2, \ldots, T$:

$$\widetilde{B}_{t',j}^{(i,t)} = \sum_{k=1}^{N^{(i)}} \widetilde{A}_{jk}^{(i)} \widetilde{L}_{t'+1,k}^{(i)} \widetilde{B}_{t'+1,k}^{(i,t)} \qquad t' = 1, 2, \ldots, t \qquad \widetilde{B}_{t,j}^{(i,t)} = 1 \tag{6.5.7}$$

$$\widetilde{F}_{t',j}^{(i,t)} = \sum_{k=1}^{N^{(i)}} \widetilde{A}_{kj}^{(i)} \widetilde{L}_{t',j}^{(i)} \widetilde{F}_{t'-1,k}^{(i,t)} \qquad t = t+1, t+2, \ldots, T \qquad \widetilde{F}_{t,k}^{(i,t)} = \widetilde{\pi}_k^{(i,0)} \tag{6.5.8}$$

where

$$\widetilde{L}_{t,j}^{(i)} = \mathbb{E}_{q(\theta^{(i,j)})}\left[\ln p(y_t|\theta^{(i,j)})\right] \qquad \widetilde{\pi}^{(i,j)} = \mathbb{E}_{q(\pi)}\left[\ln \pi^{(i,j)}\right] \tag{6.5.9}$$

and where $\widetilde{A}^{(i)}$ is a matrix with its $k$th row as $\widetilde{\pi}^{(i,k)}$. Then we can write the overall message recursions as

$$\widetilde{B}^*_{t,i} = \sum_{d=1}^{T-1-t} \widetilde{B}_{t+d,i} \widetilde{D}_{d,i} \left( \sum_{\ell=1}^{N^{(i)}} \widetilde{F}^{(i,t)}_{t+d,\ell} \right) \qquad \widetilde{B}_{t,i} = \sum_{j=1}^{N} \widetilde{A}_{ij} \widetilde{B}^*_{t,j} \tag{6.5.10}$$

$$\widetilde{F}_{t,i} = \sum_{d=1}^{T-t-1} \widetilde{F}^*_{t-d,i} \widetilde{D}_{d,i} \left( \sum_{\ell=1}^{N^{(i)}} \widetilde{B}^{(i,t)}_{t-d,\ell} \widetilde{\pi}^{(i,0)}_{\ell} \right) \qquad \widetilde{F}^*_{t,i} = \sum_{j=1}^{N} \widetilde{A}_{ji} \widetilde{F}_{t,j} \tag{6.5.11}$$

where

$$\widetilde{D}_{d,i} = \mathbb{E}_{q(\vartheta^{(i)})} \left[ \ln p(d|\vartheta^{(i)}) \right] \qquad Z = \sum_{i=1}^{N} F_{T,i} \tag{6.5.12}$$

and where $\widetilde{A}$ is a matrix with its $i$th row as $\widetilde{\pi}^{(i)}$. As in Section 6.4.2, these messages can be computed in time $\mathcal{O}(T^2 N N_{\text{sub}}^2 + TN^2)$.

Next, we calculate expected statistics in terms of these messages. To simplify notation, we write the event $\{x_{t:t+d-1} = i, x_{t-1} \neq x_t \neq x_{t+d}\}$ simply as $\{x_{t:t+d-1} = i\}$. First, note that we can write

$$E_{q(x_{1:T}, \bar{x}_{1:T})} \left[ \mathbb{I}[x_{t:t+d-1} = i] \right] = \left( \widetilde{F}^*_{t,i} \widetilde{B}_{t+d-1,i} \widetilde{D}_{d,i} \sum_{\ell=1}^{N^{(i)}} \widetilde{F}^{(i,t)}_{t,\ell} \widetilde{B}^{(i,t+d)}_{t,\ell} \right) / Z. \tag{6.5.13}$$

This decomposition follows from the definitions of the HSMM messages. Using the definition of the sub-HMM messages, we can similarly write

$$\mathbb{E}_{q(x_{1:T}, \bar{x}_{1:T})} \left[ \mathbb{I}[\bar{x}_{t'} = j, \bar{x}_{t'+1} = k] \middle| x_{t:t+d-1} = i \right] = \frac{\left( \widetilde{F}^{(i,t)}_{t',j} \widetilde{B}^{(i,t+d)}_{t'+1,k} \widetilde{L}^{(i)}_{t'+1,k} \widetilde{A}^{(i)}_{j,k} \right)}{\sum_{\ell=1}^{N^{(i)}} \widetilde{F}^{(i,t)}_{t,\ell} \widetilde{B}^{(i,t+d)}_{t,\ell}} \tag{6.5.14}$$

for any $t' = t, t+1, \ldots, t+d-2$. To compute the expected statistics, we compose these two expressions and use the basic identity that for any random variable $X$ we have

$$\mathbb{E}\left[\mathbb{I}[X \in A]\, \mathbb{I}[X \in B]\right] = \mathbb{P}\left[X \in A, X, \in B\right] \tag{6.5.15}$$

$$= \mathbb{P}\left[X \in A\right] \mathbb{P}\left[X \in B | X \in A\right] \tag{6.5.16}$$

$$= \mathbb{E}\left[\mathbb{I}[X \in A]\right]\ \mathbb{E}\left[\mathbb{I}[X \in B] \middle|\ X \in A\right]. \tag{6.5.17}$$

Therefore we can compute the expected statistics for each sub-HDP-HMM factor as

$$\hat{t}_y^{(i,j)} \triangleq \mathbb{E}_{q(x_{1:T},\bar{x}_{1:T})}\left[\sum_{t=1}^{T-1}\sum_{d=1}^{T-1-t}\mathbb{I}\left[x_{t:t+d-1}=i\right]\sum_{t'=t}^{t+d-1}\mathbb{I}\left[\bar{x}_{t'}=j\right](t_y^{(i,j)},1)\right]$$

$$=\sum_{t=1}^{T-1}\sum_{d=1}^{T-1-t}\left(\widetilde{F}_{t,i}^{*}\widetilde{B}_{t+d-1,i}\widetilde{D}_{d,i}\right)\left(\sum_{t'=t}^{t+d-1}\widetilde{F}_{t',j}^{(i,t)}\widetilde{B}_{t',j}^{(i,t+d)}\widetilde{L}_{t',j}^{(i)}\right)/Z \tag{6.5.18}$$

$$(\hat{t}_{\text{subtr}}^{(i,j)})_k \triangleq \mathbb{E}_{q(x_{1:T},\bar{x}_{1:T})}\left[\sum_{t=1}^{T-1}\sum_{d=1}^{T-1-t}\mathbb{I}\left[x_{t:t+d-1}=i\right]\sum_{t'=t}^{t+d-2}\mathbb{I}\left[\bar{x}_{t'}=j,\bar{x}_{t'+1}=k\right]\right]$$

$$=\sum_{t=1}^{T-1}\sum_{d=1}^{T-1-t}\left(\widetilde{F}_{t,i}^{*}\widetilde{B}_{t+d-1,i}\widetilde{D}_{d,i}\right)\left(\sum_{t'=t}^{t+d-2}\widetilde{F}_{t',j}^{(i,t)}\widetilde{B}_{t'+1,k}^{(i,t+d)}\widetilde{L}_{t'+1,k}^{(i)}\widetilde{A}_{j,k}^{(i)}\right)/Z \tag{6.5.19}$$

$$(\hat{t}_{\text{subinit}}^{(i)})_j \triangleq \mathbb{E}_{q(x_{1:T},x_{1:T})}\left[\sum_{t=1}^{T-1}\sum_{d=1}^{T-1-t}\mathbb{I}\left[x_{t:t+d-1}=i\right]\mathbb{I}\left[\bar{x}_t=j\right]\right]$$

$$=\sum_{t=1}^{T-1}\sum_{d=1}^{T-1-t}\left(\widetilde{F}_{t,i}^{*}\widetilde{B}_{t+d-1,i}\widetilde{D}_{d,i}\right)\left(\widetilde{F}_{t,j}^{(i,t)}\widetilde{B}_{t,j}^{(i,t+d)}\right)/Z \tag{6.5.20}$$

Furthermore, we can compute the expected statistics for each HDP-HSMM factor as

$$(\hat{t}_{\text{dur}}^{(i)})_d \triangleq \mathbb{E}_{q(x_{1:T})}\left[\sum_{t=1}^{T-1}\mathbb{I}\left[x_{t:t+d}=i\right]\right]$$

$$=\sum_{t=1}^{T-1}\widetilde{F}_{t,i}^{*}\widetilde{B}_{t+d,i}\widetilde{D}_{d,i}\left(\sum_{\ell=1}^{N^{(i)}}\widetilde{F}_{t,\ell}^{(i,t)}\widetilde{B}_{t,\ell}^{(i,t+d)}\right)/Z \tag{6.5.21}$$

$$(\hat{t}_{\text{trans}}^{(i)})_j \triangleq \mathbb{E}_{q(x_{1:T})}\left[\sum_{t=1}^{T-1}\mathbb{I}\left[x_t=i,x_{t+1}=j\right]\right]=\sum_{t=1}^{T-1}\widetilde{F}_{t,i}\widetilde{B}_{t,j}^{*}\widetilde{A}_{i,j}/Z \tag{6.5.22}$$

$$(\hat{t}_{\text{init}})_i \triangleq \mathbb{E}_{q(x_{1:T})}\mathbb{I}[x_1=i]=\widetilde{F}_{1,i}^{*}\widetilde{B}_{1,i}/Z \tag{6.5.23}$$

While these expected statistics expressions appear complex when fully expanded, the expressions are in fact quite modular: each involves the expectation of an HSMM segment indicator, which is computed using the HSMM messages, and possibly an expectation in terms of a sub-HMM statistic, which is computed using the sub-HMM messages.

Using the notation of Chapter 5, the corresponding SVI updates to the variational factors on the model parameters are then

**Algorithm 6.1** Sub-HMM SVI

Initialize global variational parameters $\widetilde{\eta_\theta}^{(i,j)}$, $\widetilde{\alpha}^{(i,j)}$, $\widetilde{\alpha}^{(i)}$, and $\widetilde{\eta}_\vartheta^{(i)}$
**for** $t = 1, 2, \ldots$ **do**
  Sample minibatch index $\hat{k}$ uniformly from $\{1, 2, \ldots, K\}$
  Using minibatch $\bar{y}^{(\hat{k})}$, compute sub-HMM messages using (6.5.7)-(6.5.8) and HSMM messages using (6.5.10)-(6.5.11)
  Using the messages, compute $\hat{t}_y^{(i,j)}$, $\hat{t}_{\text{subtr}}^{(i,j)}$, $\hat{t}_{\text{subinit}}^{(i)}$, $\hat{t}_{\text{dur}}^{(i)}$, $\hat{t}_{\text{trans}}^{(i)}$, and $\hat{t}_{\text{init}}$ using (6.5.18)-(6.5.23).
  Update each $\widetilde{\eta_\theta}^{(i,j)}$, $\widetilde{\alpha}^{(i,j)}$, $\widetilde{\alpha}^{(i)}$, and $\widetilde{\eta}_\vartheta^{(i)}$ using (6.5.24)-(6.5.29)

$$\widetilde{\eta_\theta}^{(i,j)} \leftarrow (1-\rho)\widetilde{\eta_\theta}^{(i,j)} + \rho(\eta_\theta^{(i,j)} + s \cdot \hat{t}_y^{(i,j)}) \tag{6.5.24}$$

$$\widetilde{\alpha}^{(i,j)} \leftarrow (1-\rho)\widetilde{\alpha}^{(i,j)} + \rho(\alpha^{(i)} + s \cdot \hat{t}_{\text{subtr}}^{(i,j)}) \tag{6.5.25}$$

$$\widetilde{\alpha}^{(i,0)} \leftarrow (1-\rho)\widetilde{\alpha}^{(i,0)} + \rho(\alpha^{(i)} + s \cdot \hat{t}_{\text{subinit}}^{(i)}) \tag{6.5.26}$$

$$\widetilde{\alpha}^{(i)} \leftarrow (1-\rho)\widetilde{\alpha}^{(i)} + \rho(\alpha + s \cdot \hat{t}_{\text{trans}}^{(i)}) \tag{6.5.27}$$

$$\widetilde{\alpha}^{(0)} \leftarrow (1-\rho)\widetilde{\alpha}^{(0)} + \rho(\alpha + s \cdot \hat{t}_{\text{init}}^{(i)}) \tag{6.5.28}$$

$$\widetilde{\eta}_\vartheta^{(i)} \leftarrow (1-\rho)\widetilde{\eta}_\vartheta^{(i)} + \rho(\eta_\vartheta^{(i)} + s(\textstyle\sum_{d=1}^{T}(\hat{t}_{\text{dur}}^{(i)})_d \cdot (t_d(d), 1))). \tag{6.5.29}$$

for some stepsize $\rho$ and minibatch scaling $s$ as in Section 5.1.2. We summarize the overall algorithm in Algorithm 6.1.

## ■ 6.6 Experiments

As discussed in Section 6.1, one natural motivation for models with multiple timescales is speech analysis. Individual phonetic units, such as phonemes, have internal dynamical structure that can be modeled by an HMM with Gaussian emissions [58, 68]. At the same time, it is desirable to model the dynamical patterns among the phonemes themselves. The model and inference algorithms developed in this chapter can be applied to capture these two timescales of dynamics. Furthermore, by utilizing both explicit duration modeling and a Bayesian approach we can easily incorporate informative prior knowledge and encourage the model to learn interpretable representations.

Similar models have been applied successfully to tasks in speech analysis. In particular, Lee and Glass [68] develop a Bayesian nonparametric approach in which phonetic units are each modeled as fixed-size HMMs and the number of such units is discovered using a Dirichlet process mixture. The authors show that the discovered phonetic units are highly correlated with English phones and that the model can be used for

some speech tasks to achieve state-of-the-art performance relative to other unsupervised methods. Our model can be viewed as a refinement of this approach in two ways: first, our model admits explicit phonetic unit duration and transition modeling, and second, the inference algorithms we develop allow our model and similar models to be fit to large datasets much more efficiently. Indeed, our algorithms allow such models to be fit in minutes or hours of computation time instead of days or weeks.

In this section we describe an application of our models and algorithms to semi-supervised phonetic unit modeling based on the approach of Lee and Glass [68]. In particular, we demonstrate the advantages of using explicit duration priors, of modeling dynamics within phonetic units, and of using scalable inference algorithms.

The remainder of this section is organized as follows. In Section 6.6.1 we describe the dataset and features we use to train and evaluate the model. In Section 6.6.2 we describe a general approach to set the hyperparameters for informative duration priors. Finally, in Section 6.6.3 we describe our training procedure and experimental results.

### ■ 6.6.1 Dataset and features

Our setup follows Lee and Glass [68] closely. We use the TIMIT dataset, which consists of recordings of 630 speakers each reading 10 sentences, for a total of 6300 example sequences. We process these recordings into 13-dimensional MFCC features [21] using sliding windows of width 25ms spaced every 10ms. We concatenate the MFCCs with their first- and second-order numerical time derivatives to form a 39-dimensional feature vector. We also center and whiten these features to have zero mean and identity covariance.

The resulting dataset contains 6300 sequences of 39-dimensional features, where the sequence lengths vary from 90 to 777 with an average length of 305. See Figure 6.2 for a histogram of sequence lengths. The total number of features is 1,925,362 frames.

In addition, we follow Lee and Glass [68] and use the changepoint detector of Glass [41] to accelerate our training algorithm. We include these detected possible changepoints while training our model to reduce the complexity of the message passing computation using the methods developed for the energy disaggregation application of Chapter 3. See Figure 6.3 for a typical set of examples showing the detected changepoints and the true changepoints.

The TIMIT dataset is also fully expert-labeled with phonetic units, and we make use of a small subset of these labels to set our priors and initialize our fitting procedure, as we describe in the subsequent sections.

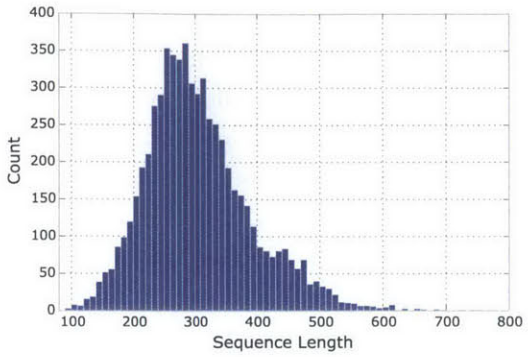



**Figure 6.2:** Histogram of TIMIT sequence lengths.

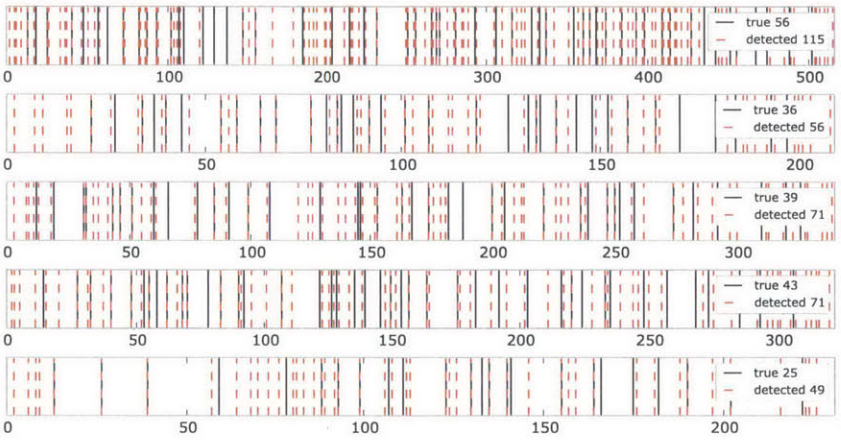
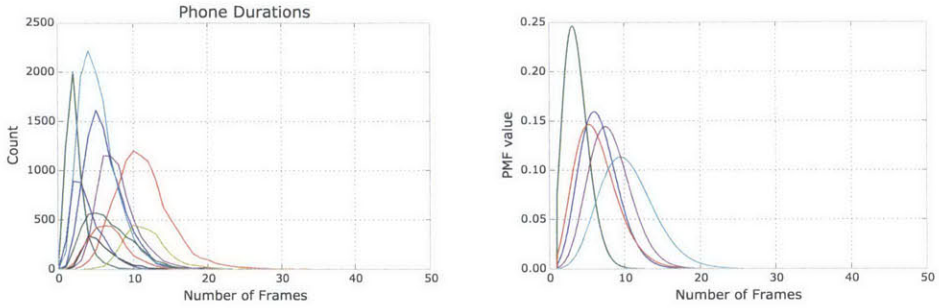



**Figure 6.3:** Detected possible changepoints

## ■ 6.6.2 Setting informative duration priors

We wish to use informative duration priors to encourage the model to learn interpretable phonetic units. In this subsection we describe a general method for setting duration hyperparameters.

Phonetic units have durations that are well-modeled by negative binomial distributions; see Figure 6.4(a) for typical phonetic unit duration distributions from the labels in the TIMIT dataset. Recall from Chapter 4 that a negative binomial distribution has parameters $(r, p)$, where $r \in \{1, 2, \ldots, r_{\max}\}$ and $0 < p < 1$. We use priors of the form




(a) Ground-truth durations (b) Prior samples

**Figure 6.4:** Phonetic unit durations in the labeled dataset and informative priors set using the method of Section 6.6.2

$p(r,p) = p(r)p(p|r)$ where

$$p(r = j|\nu) = \nu_j \qquad p(p|r = j) = \text{Beta}(a_j, b_j) \tag{6.6.1}$$

where $(\nu_j, a_j, b_j)$ for $j = 1, 2, \ldots, r_{\max}$ are the hyperparameters we wish to determine from labeled examples.

A natural way to set hyperparameters is via empirical Bayes [38], in which one chooses hyperparameters to maximize the likelihood of an observed training set. While we have no training set of $(r,p)$ parameter pairs available for such a procedure, we can simulate an appropriate set of parameters by using the Gibbs sampling procedure developed in Section 4.4.2 and some labeled durations. Using a set of durations $\{d_i\}_{i=1}^S$ drawn from the expert labels in the TIMIT dataset, we collect samples of $(r,p)$ pairs from $p(r,p|\{d_i\}, \nu^0, a^0, b^0)$, where $\nu_j^0 = \frac{1}{r_{\max}}$ and $a_j^0 = b_j^0 = 1$ for $j = 1, 2, \ldots, r_{\max}$ are chosen to be non-informative. Using these simulated samples $\{(\hat{r}_k, \hat{p}_k)\}_{k=1}^K$, we then choose hyperparameters via maximum likelihood. We choose $S$, the number of duration examples used to set the hyperparameters, to correspond to 2.5% of the labeled examples, and we set $K$, the number of simulated samples, to be equal to $S$.

See Figure 6.4(b) for samples drawn from this prior. By comparing these duration distribution samples to the histograms in Figure 6.4(a), it is clear that the prior summarizes the empirical distribution over typical phoneme duration means and variances well. In addition, the negative binomial duration distribution class is able to represent the empirical phoneme duration distributions, which look substantially different from the geometric durations to which we would be restricted with a purely HMM-based approach.

**Table 6.1:** Fit times and per-frame predictive likelihoods

| | Sub-HMM | Sub-GMM | HDP-HMM |
|---|---|---|---|
| Pred. Like. (nats) | -44.046 | -47.599 | -47.940 |
| Fit Time (min.) | 110 | 67 | 10 |

### 6.6.3 Experimental procedure and results

In this subsection we fit three alternative models, two of which are developed in this thesis, and compare their performance both at prediction and on a phonetic unit segmentation task. First, we fit the nonparametric model developed in this chapter, which we refer to as the HDP-HSMM Sub-HMM model. Second, we fit an HDP-HSMM with Gaussian mixture model (GMM) emissions, which we refer to as the HDP-HSMM Sub-GMM model. This second model is different from the first only in that by using GMM emissions instead of sub-HMM emissions, the internal structure of the phonetic units is not modeled. Finally, we fit an HDP-HMM for comparison. The HDP-HMM does not include the duration prior that the other two models can incorporate. Each model has 39-dimensional Gaussian emission distributions, with Normal Inverse-Wishart priors with hyperparameters set as $\mu_0 = 0$, $\Sigma_0 = I$, $\kappa_0 = 0.5$, and $\nu_0 = 45$. For each of the three models, we are able to scale efficient inference to this large dataset using the algorithms developed in this chapter and in Chapter 5, allowing the models to be fit orders of magnitude faster than previous methods.

Often in speech analysis there is an abundance of unlabeled data but only a very limited amount of labeled data. In such settings, labeled data is used to set priors and initializations, while an unsupervised inference procedure is used with the large amount of unlabeled data. Accordingly, we use the labels from 2.5% of the full dataset to set our prior hyperparameters using Gibbs sampling. We also use this small subset of labeled data to initialize our inference procedure. We perform inference over the unlabeled data in a single pass over the dataset using a minibatch size of 50 sequences and a stepsize sequence $\rho^{(t)} = (t + \tau)^{-\kappa}$ where we chose $\tau = 0$ and $\kappa = 0.6$.

Table 6.1 summarizes both the fitting runtimes and the predictive performance of each model. We measure predictive performance by computing the per-frame predictive likelihood on 20 held-out sequences, where a larger value indicates a higher average likelihood assigned to each held-out frame and hence better predictions. The per-frame predictive likelihoods are very similar, indicating that the alternative models perform comparably well on predictive measures. However, their predictive performance does not give any insight into the interpretability of the latent structure learned, which we discuss next.

To evaluate the quality and interpretability of the learned latent parameters, we

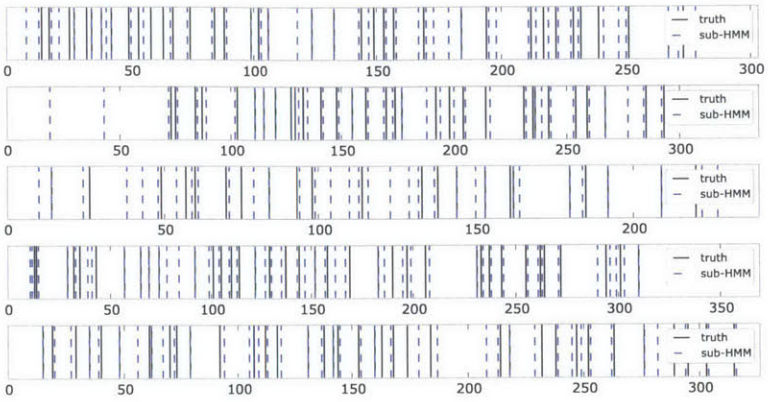
**Table 6.2:** Error rates for the segmentation task

| | Sub-HMM | Sub-GMM | HDP-HMM |
|---|---|---|---|
| Missed Detections | 22.0 | 21.9 | 24 |
| False Positives | 31.9 | 35.0 | 59.8 |

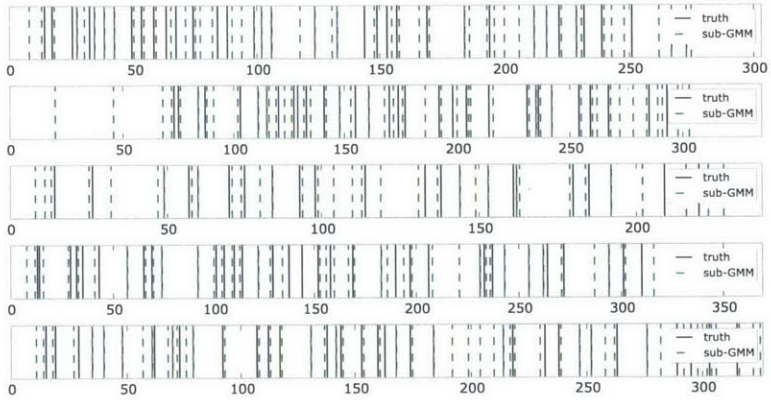
consider a segmentation task similar to the one considered by Lee and Glass [68]. On the 20 held-out sequences and using no changepoint information from the changepoint detector, we compute the optimal variational factor over the label sequence (or state sequence in the case of the HDP-HMM) and then perform a Viterbi decoding to find the most probable joint assignment according to that variational factor. Finding this most probable label sequence (or state sequence) assignment evaluates each model's ability to discover modes that correspond to phonemes, where the HDP-HMM is unable to distinguish the dynamics at multiple timescales present in the data. We then compare the changepoints in the Viterbi sequence to the true changepoints and measure both the missed detection and false positive error rates. Following Lee and Glass [68] and Scharenborg et al. [100], we allow a 20ms tolerance window to compute detections.

We summarize the segmentation performance of the three models in Table 6.2. We find that both of the models which include explicit duration modeling perform significantly better than the HDP-HMM at both missed detection and false positive error rates. In addition, we find that modeling the dynamics within each phonetic unit with the Sub-HMM model further reduces the false positive rate. The HDP-HMM, which cannot separate timescales because it lacks explicit duration modeling, tends to over-segment relative to the intended phonetic unit segmentation, leading to a very high false positive error rate. The Sub-HMM changepoints also perform well qualitatively; in Figure 6.5 we show 5 typical examples of the changepoints detected by each model.

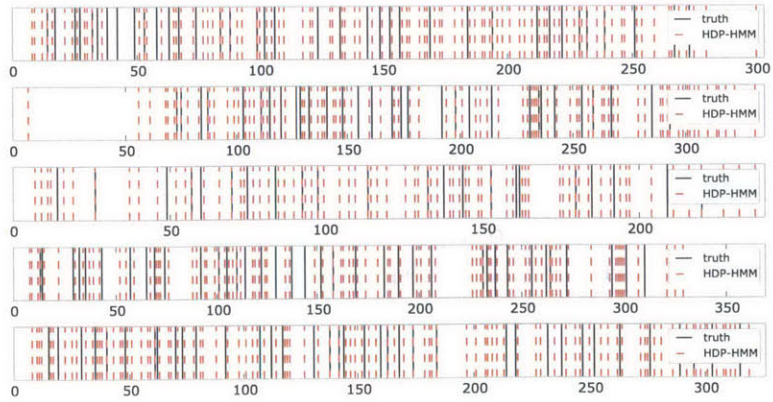
These experiments demonstrate advantages of both our algorithms and our models. With our SVI algorithms we are able to perform inference in a single pass over the dataset in the time it would require to compute a single Gibbs sampling or batch mean field update. Thus our algorithms allow inference in each of these models to scale to large datasets efficiently, reducing the computation time by orders of magnitude and enabling even larger datasets to be explored. By comparing three related models, we also show that explicit duration modeling provides a significant boost to segmentation performance, with the Sub-HMM model refinement providing a further increase in performance. These models and algorithms may provide new tools for speech researchers to analyze detailed structure while imposing model regularities with interpretable prior information.




**(a)** Detected by the Sub-HMM model




**(b)** Detected by the Sub-GMM model




**(c)** Detected by the HDP-HMM model

**Figure 6.5:** Phonetic unit boundaries detected by the three models.

## 6.7 Conclusion

This chapter composes the ideas of Chapters 3, 4, and 5 to develop both new models and new efficient and scalable algorithms. In particular, it shows that the ideas developed in this thesis can be readily extended. The flexible Bayesian nonparametric approach to modeling dynamics at multiple timescales may provide new insights into complex phenomena, and the algorithms we develop enable such rich models to be fit to large enough datasets. Finally, our speech application shows the promise and potential utility of explicit duration modeling in a Bayesian framework, both for performance and for interpretability.

Chapter 7

# Analyzing Hogwild Parallel Gaussian Gibbs Sampling

## 7.1 Introduction

Scaling probabilistic inference algorithms to large datasets and parallel computing architectures is a challenge of great importance and considerable current research interest, and great strides have been made in designing parallelizeable algorithms. Along with the powerful and sometimes complex new algorithms, a very simple strategy has proven to be surprisingly useful in some situations: running local Gibbs sampling updates on multiple processors in parallel while only periodically communicating updated statistics (see Section 7.4 for details). We refer to this strategy as "Hogwild Gibbs sampling" in reference to recent work [84] in which sequential computations for computing gradient steps were applied in parallel (without global coordination) to great beneficial effect.

This Hogwild Gibbs sampling strategy is not new; indeed, Gonzalez et al. [42] attributes a version of it to the original Gibbs sampling paper (see Section 7.2 for a discussion), though it has mainly been used as a heuristic method or initialization procedure without theoretical analysis or guarantees. However, extensive empirical work on Approximate Distributed Latent Dirichlet Allocation (AD-LDA) [83, 82, 73, 7, 55], which applies the strategy to generate samples from a collapsed LDA model [12], has demonstrated its effectiveness in sampling LDA models with the same or better predictive performance as those generated by standard serial Gibbs [83, Figure 3]. The results are empirical and so it is difficult to understand how model properties and algorithm parameters might affect performance, or whether similar success can be expected for any other models. There have been recent advances in understanding some of the particular structure of AD-LDA [55], but a thorough theoretical explanation for the effectiveness and limitations of Hogwild Gibbs sampling is far from complete.

Sampling-based inference algorithms for complex Bayesian models have notoriously resisted theoretical analysis, so to begin an analysis of Hogwild Gibbs sampling we consider a restricted class of models that is especially tractable for analysis: Gaussians.

Gaussian distributions and algorithms are tractable because of their deep connection with linear algebra. Further, Gaussian sampling is of significant interest in its own right, and there is active research in developing effective Gaussian samplers [72, 89, 90, 29]. Gaussian Hogwild Gibbs sampling can be used in conjunction with those methods to allow greater parallelization and scalability, provided some understanding of its applicability and tradeoffs.

The main contribution of this chapter is a linear algebraic framework for analyzing the stability and errors in Gaussian Hogwild Gibbs sampling. Our framework yields several results, including a simple proof for a sufficient condition for all Gaussian Hogwild Gibbs sampling processes to be stable and yield the correct asymptotic mean no matter the allocation of variables to processors. Our framework also provides an analysis of errors introduced in the process covariance, which in one case of interest leads to an inexpensive correction for those errors.

In Section 7.2 we discuss some related work in greater detail. In Section 7.3 we overview known connections between Gaussian sampling and linear system solvers, connections on which we build to provide an analysis for Hogwild Gibbs sampling. In Section 7.4 we precisely define the parallel updating scheme. Finally, in Section 7.5 we present our analytical framework and main results on Gaussian models.

## ■ 7.2 Related work

There has been significant work on constructing parallel Gibbs sampling algorithms, and the contributions are too numerous to list here. One recent body of work [42] provides exact parallel Gibbs samplers which exploit particular graphical model structure for parallelism. The algorithms are supported by the standard Gibbs sampling analysis, and the authors point out that while heuristic parallel samplers such as the AD-LDA sampler offer easier implementation and often greater parallelism, they are currently not supported by much theoretical analysis. Gonzalez et al. [42] attribute one version (see Section 7.4) of Hogwild Gibbs to the original Gibbs sampling paper [39] and refer to it as Synchronous Gibbs, though the Gibbs sampling paper only directly discusses an asynchronous implementation of their exact Gibbs sampling scheme rather than a parallelized approximation [39, Section XI]. Gonzalez et al. [42] also gives a result on Synchronous Gibbs in the special case of two processors.

The parallel sampling work that is most relevant to the proposed Hogwild Gibbs sampling analysis is the thorough empirical demonstration of AD-LDA [83, 82, 73, 7, 55] and its extensions. The AD-LDA sampling algorithm is an instance of the strategy we have named Hogwild Gibbs, and Bekkerman et al. [7, Chapter 11] suggests applying the strategy to other latent variable models.

The work of Ihler and Newman [55] provides some understanding of the effective-

ness of a variant of AD-LDA by bounding in terms of run-time quantities the one-step error probability induced by proceeding with sampling steps in parallel, thereby allowing an AD-LDA user to inspect the computed error bound after inference [55, Section 4.2]. In experiments, the authors empirically demonstrate very small upper bounds on these one-step error probabilities, e.g. a value of their parameter $\varepsilon = 10^{-4}$ meaning that at least 99.99% of samples are expected to be drawn just as if they were sampled sequentially. However, this per-sample error does not necessarily provide a direct understanding of the effectiveness of the overall algorithm because errors might accumulate over sampling steps; indeed, understanding this potential error accumulation is of critical importance in iterative systems. Furthermore, the bound is in terms of empirical run-time quantities, and thus it does not provide guidance regarding on which other models the Hogwild strategy may be effective. Ihler and Newman [55, Section 4.3] also provides approximate scaling analysis by estimating the order of the one-step bound in terms of a Gaussian approximation and some distributional assumptions.

Finally, Niu et al. [84] provides both a motivation for Hogwild Gibbs sampling as well as the Hogwild name. The authors present "a lock-free approach to parallelizing stochastic gradient descent" (SGD) by providing analysis that shows, for certain common problem structures, that the locking and synchronization needed for a stochastic gradient descent algorithm to converge on a multicore architecture are unnecessary, and in fact the robustness of the SGD algorithm compensates for the uncertainty introduced by allowing processors to perform updates without locking their shared memory.

## ■ 7.3 Gaussian sampling background

In this section we fix notation for Gaussian distributions and describe known connections between Gaussian sampling and a class of stationary iterative linear system solvers which are useful in analyzing the behavior of Hogwild Gibbs sampling.

The density of a Gaussian distribution on $n$ variables with mean vector $\mu$ and positive definite[1] covariance matrix $\Sigma \succ 0$ has the form

$$p(x) \propto \exp\left\{-\tfrac{1}{2}(x-\mu)^{\mathsf{T}}\Sigma^{-1}(x-\mu)\right\} \propto \exp\left\{-\tfrac{1}{2}x^{\mathsf{T}}Jx + h^{\mathsf{T}}x\right\} \tag{7.3.1}$$

where we have written the information parameters $J \triangleq \Sigma^{-1}$ and $h \triangleq J\mu$. The matrix $J$ is often called the *precision matrix* or *information matrix*, and it has a natural interpretation in the context of Gaussian graphical models: its entries are the coefficients on pairwise log potentials and its sparsity pattern is exactly the sparsity pattern of a graphical model. Similarly $h$, also called the *potential vector*, encodes node potentials and evidence.

[1] We assume models are non-degenerate, i.e. that covariances are of full rank.

In many problems [113] one has access to the pair $(J, h)$ and must compute or estimate the moment parameters $\mu$ and $\Sigma$ (or just the diagonal) or generate samples from $\mathcal{N}(\mu, \Sigma)$. Sampling provides both a means for estimating the moment parameters and a subroutine for other algorithms. Computing $\mu$ from $(J, h)$ is equivalent to solving the linear system $J\mu = h$ for $\mu$.

One way to generate samples is via Gibbs sampling, in which one iterates sampling each $x_i$ conditioned on all other variables to construct a Markov chain for which the invariant distribution is the target $\mathcal{N}(\mu, \Sigma)$. The conditional distributions for Gibbs sampling steps are

$$p(x_i|x_{\neg i} = \bar{x}_{\neg i}) \propto \exp\left\{-\frac{1}{2}\begin{pmatrix} x_i & \bar{x}_{\neg i}^{\mathsf{T}}\end{pmatrix}\begin{pmatrix} J_{ii} & J_{i\neg i} \\ J_{\neg ii} & J_{\neg i\neg i}\end{pmatrix}\begin{pmatrix} x_i \\ \bar{x}_{\neg i}\end{pmatrix} - \begin{pmatrix} h_i & h_{\neg i}^{\mathsf{T}}\end{pmatrix}\begin{pmatrix} x_i \\ \bar{x}_{\neg i}\end{pmatrix}\right\} \tag{7.3.2}$$

$$\propto \exp\left\{-\frac{1}{2}J_{ii}x_i^2 + (h_i - J_{i\neg i}\bar{x}_{\neg i})x_i\right\} \tag{7.3.3}$$

where the indexing $x_{\neg i} \triangleq (x_j : j \neq i) \in \mathbb{R}^{n-1}$ denotes all the variables other than $x_i$ and $J_{i\neg i} \triangleq (J_{ij} : j \neq i)$ denotes the $i$th row of $J$ with its $i$th entry removed. That is, we update each $x_i$ to be a scalar Gaussian sample with mean $\frac{1}{J_{ii}}(h_i - J_{i\neg i}\bar{x}_i)$ and variance $\frac{1}{J_{ii}}$ or, equivalently,

$$x_i \leftarrow \frac{1}{J_{ii}}(h_i - J_{i\neg i}\bar{x}_{\neg i}) + v_i \quad \text{where} \quad v_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \frac{1}{J_{ii}}). \tag{7.3.4}$$

Since each variable update is a linear function of other variables with added Gaussian noise, we can collect one scan for $i = 1, 2, \ldots, n$ into a matrix equation relating the sampler state vector at $t$ and $t + 1$:

$$x^{(t+1)} = -D^{-1}Lx^{(t+1)} - D^{-1}L^{\mathsf{T}}x^{(t)} + D^{-1}h + D^{-\frac{1}{2}}\tilde{v}^{(t)} \tag{7.3.5}$$

$$\tilde{v}^{(t)} \overset{\text{iid}}{\sim} \mathcal{N}(0, I). \tag{7.3.6}$$

where we have split $J = L + D + L^{\mathsf{T}}$ into its strictly lower-triangular, diagonal, and strictly upper-triangular parts, respectively. Note that $x^{(t+1)}$ appears on both sides of the equation, and that the sparsity patterns of $L$ and $L^{\mathsf{T}}$ ensure that the updated value $x_i^{(t+1)}$ depends only on $x_a^{(t)}$ and $x_b^{(t+1)}$ for all $a > i$ and $b < i$. We can rearrange the equation into an update expression:

$$(I + D^{-1}L)x^{(t+1)} = -D^{-1}L^{\mathsf{T}}x^{(t)} + D^{-1}h + D^{-\frac{1}{2}}v^{(t)} \tag{7.3.7}$$

$$x^{(t+1)} = -(D+L)^{-1}L^{\mathsf{T}}x^{(t)} + (D+L)^{-1}h + (D+L)^{-1}D^{\frac{1}{2}}v^{(t)} \tag{7.3.8}$$

$$= -(D+L)^{-1}L^{\mathsf{T}}x^{(t)} + (D+L)^{-1}h + (D+L)^{-1}\tilde{v}^{(t)} \tag{7.3.9}$$

$$\tilde{v}^{(t)} \overset{\text{iid}}{\sim} \mathcal{N}(0, D). \tag{7.3.10}$$

The expectation of this update is exactly the Gauss-Seidel iterative linear system solver update [9, Section 7.3] applied to $J\mu = h$, i.e. $x^{(t+1)} = -(D+L)^{-1}L^{\mathsf{T}}x^{(t)} + (D+L)^{-1}h$. Therefore a Gaussian Gibbs sampling process can be interpreted as Gauss-Seidel iterates on the system $J\mu = h$ with appropriately-shaped noise injected at each iteration.

Gauss-Seidel is one instance of a stationary iterative linear solver based on a *matrix splitting*. In general, one can construct a stationary iterative linear solver for any splitting $J = M - N$ where $M$ is invertible, and similarly one can construct iterative Gaussian samplers via

$$x^{(t+1)} = (M^{-1}N)x^{(t)} + M^{-1}h + M^{-1}v^{(t)} \tag{7.3.11}$$

$$v^{(t)} \overset{\text{iid}}{\sim} \mathcal{N}(0, M^{\mathsf{T}} + N) \tag{7.3.12}$$

with the constraint that $M^{\mathsf{T}} + N \succeq 0$ (i.e. that the splitting is P-regular [77]). For a stationary iterative process like (7.3.11) to be *stable* or *convergent* for any initialization we require the eigenvalues of its update map to lie in the interior of the complex unit disk, i.e. $\rho(M^{-1}N) \triangleq \max_i |\lambda_i(M^{-1}N)| < 1$ [9, Lemma 7.3.6]. The Gauss-Seidel solver (and Gibbs sampling) correspond to choosing $M$ to be the lower-triangular part of $J$ and $N$ to be the negative of the strict upper-triangle of $J$. $J \succ 0$ is a sufficient condition for Gauss-Seidel to be convergent [9, Theorem 7.5.41] [101], and the connection to Gibbs sampling provides an alternative proof.

For solving linear systems with splitting-based algorithms, the complexity of solving linear systems in $M$ directly affects the computational cost per iteration. For the Gauss-Seidel splitting (and hence Gibbs sampling), $M$ is chosen to be lower-triangular so that the corresponding linear system can be solved efficiently via back-substitution. In the sampling context, the per-iteration computational complexity is also determined by the covariance of the injected noise process $v^{(t)}$, because at each iteration one must sample from a Gaussian distribution with covariance $M^{\mathsf{T}} + N$.

We highlight one other standard stationary iterative linear solver that is relevant to analyzing Gaussian Hogwild Gibbs sampling: Jacobi iterations, in which one splits

$J = D - A$ where $D$ is the diagonal part of $J$ and $A$ is the negative of the off-diagonal part. Due to the choice of a diagonal $M$, each coordinate update depends only on the previous sweep's output, and thus the Jacobi update sweep can be performed in parallel. A sufficient condition for the convergence of Jacobi iterates is for $J$ to be a generalized diagonally dominant matrix (i.e. an H-matrix) [9, Definition 5.13]. A simple proof [2] due to Ruozzi and Tatikonda [96], is to consider Gauss-Seidel iterations on a *lifted* $2n \times 2n$ system:

$$\begin{pmatrix} D & -A \\ -A & D \end{pmatrix} \xrightarrow{\text{G-S update}} \begin{pmatrix} D^{-1} & \\ D^{-1}AD^{-1} & D^{-1} \end{pmatrix} \begin{pmatrix} & A \\ & \end{pmatrix} = \begin{pmatrix} & D^{-1}A \\ & (D^{-1}A)^2 \end{pmatrix} \tag{7.3.13}$$

where zero entries are left blank where dimensions can be inferred. Therefore one iteration of Gauss-Seidel on the lifted system corresponds to two iterations of the Jacobi update $D^{-1}A$ to the latter $n$ entries in the lifted system, so Jacobi iterations converge if Gauss-Seidel on the lifted system converges. Furthermore, a sufficient condition for Gauss-Seidel to converge on the lifted system is for the lifted matrix to be positive definite, and by taking Schur complements we require $D - AD^{-1}A \succ 0$ or $I - (D^{-\frac{1}{2}}AD^{-\frac{1}{2}})(D^{-\frac{1}{2}}AD^{-\frac{1}{2}}) \succ 0$, which is equivalent to requiring strict generalized diagonal dominance of $J$ [9, Theorem 5.14].

## ■ 7.4 Hogwild Gibbs model

In this section, we define the Hogwild Gibbs computational model and fix some notation for the iterative process that we use for the remainder of the chapter.

As with standard Gibbs sampling, we assume we are given a collection of $n$ random variables $\{x_i : i \in [n]\}$ where $[n] \triangleq \{1, 2, \ldots, n\}$ and that we can sample from the conditional distributions $x_i | x_{\neg i}$. Gibbs sampling is an iterative Markov process on the *state vector* $x^{(t)}$ for times $t = 1, 2, \ldots$ so that the stationary distribution is the joint distribution of $\{x_i : i \in [n]\}$.

For Hogwild Gibbs, we assume we are given a partition $\{\mathcal{I}_1, \mathcal{I}_1, \ldots, \mathcal{I}_K\}$ of $[n]$ that represents an allocation of the state vector to $K$ processors, so that the $k$th processor updates the state values indexed by $\mathcal{I}_k$. We assume each partition element $\mathcal{I}_k$ is contiguous and ordered and we write $x_{\mathcal{I}_k} \triangleq (x_i : i \in \mathcal{I}_k)$ to denote the corresponding sub-vector of any vector $x$. We keep this partition fixed over time for the majority of this chapter, though we describe a generalization in Theorem 7.6.2.

The Hogwild Gibbs algorithm is shown in Algorithm 7.1. We define two iterations: *outer iterations*, which count the number of global synchronizations among the proces-

[2] When $J$ is symmetric one can arrive at the same condition by applying a similarity transform as in Proposition 7.7.3. We use the lifting argument here because we extend the idea in our other proofs.

**Algorithm 7.1** Hogwild Gibbs

**Input:** Joint distribution over $x = (x_1, \ldots, x_n)$, partition $\{\mathcal{I}_1, \ldots, \mathcal{I}_K\}$ of $\{1, 2, \ldots, n\}$
Initialize $\bar{x}^{(1)}$
**for** $t = 1, 2, \ldots$ **do**
  **for** $k = 1, 2, \ldots, K$ in parallel **do**
    $\bar{x}_{\mathcal{I}_k}^{(t+1)} \leftarrow \text{LocalGibbs}(\bar{x}^{(t)}, \mathcal{I}_k, q(t, k))$

**function** LocalGibbs($\bar{x}$, $\mathcal{I}$, $q$)
  **for** $j = 1, 2, \ldots, q$ **do**
    **for** $i \in \mathcal{I}$ in order **do**
      $\bar{x}_i \leftarrow$ sample $x_i | x_{\neg i} = \bar{x}_{\neg i}$
  **return** $\bar{x}$

sors, and *inner iterations*, which count processor-local Gibbs scans. That is, during outer iteration $t$ (for each $t = 1, 2, \ldots$), processor $k$ runs a number $q(t, k)$ of inner iterations, each of which consists of a systematic scan Gibbs update [94, Algorithm A.40] of its variables indexed by $\mathcal{I}_k$. During the inner iterations on each processor, the processors do not communicate; in particular, all inner iterations on processor $k$ compute Gibbs updates using out-of-date values of $x_i$ for $i \notin \mathcal{I}_k$. Processors synchronize values once per outer iteration, and we write $x^{(t)}$ for the globally shared value before the inner iterations of outer iteration $t$. For the majority of this chapter, we fix the number of inner iterations performed to be constant for all processors and for all outer iterations, so that $q(t, k) = q$, though we describe a generalization in Theorem 7.6.2.

There are several special cases of this general scheme that may be of interest. The Synchronous Gibbs scheme of Gonzalez et al. [42] corresponds to associating one variable to each processor, so that $|\mathcal{I}_k| = 1$ for each $k = 1, 2, \ldots, K$ (in which case we may take $q = 1$ since no local iterations are needed with a single variable). More generally, it is particularly interesting to consider the case where the partition is arbitrary and $q$ is very large, in which case the local Gibbs iterations can mix and exact block samples are drawn on each processor using old statistics from other processors for each outer iteration. Finally, note that setting $K = 1$ and $q = 1$ reduces to standard Gibbs sampling on a single processor.

## ■ 7.5 Gaussian analysis setup

Given that Gibbs sampling iterations and Jacobi solver iterations can each be written as iterations of a stochastic linear dynamical system (LDS), it is not surprising that Gaussian Hogwild Gibbs sampling can also be expressed as an LDS by appropriately composing these ideas. In this section we describe the LDS corresponding to Gaussian Hogwild Gibbs sampling and provide convergence and error analysis, along with a

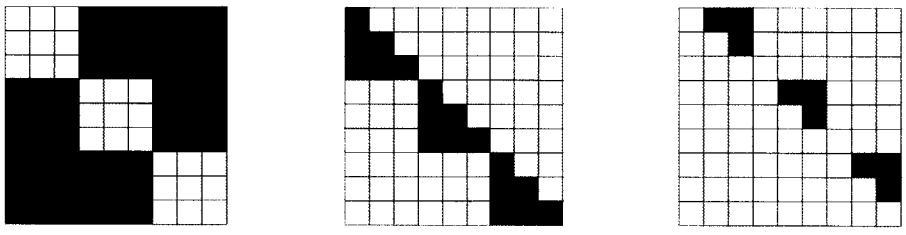



**Figure 7.1:** Support pattern (in black) of the Hogwild splitting $J = B - C - A$ with $n = 9$ and the processor partition $\{\{1,2,3\},\{4,5,6\},\{7,8,9\}\}$.

connection to a class of linear solvers.

Given a joint Gaussian distribution of dimension $n$ represented by a pair $(J, h)$ as in (7.3.1), consider a block-Jacobi splitting of $J$ into its block diagonal and off-block-diagonal components, $J = D_{\text{bd}} - A$, according to the partition. $A$ includes the entries of $J$ corresponding to cross-processor terms, and this block-Jacobi splitting will model the outer iterations in Algorithm 7.1. We further perform a Gauss-Seidel splitting on $D_{\text{bd}}$ into (block-diagonal) lower-triangular and strictly upper-triangular parts, $D_{\text{bd}} = B - C$; these processor-local Gauss-Seidel splittings model the inner itertions in Algorithm 7.1. We refer to this splitting $J = B - C - A$ as the Hogwild splitting; see Figure 7.1 for an example.

For each outer iteration of the Hogwild Gibbs sampler we perform $q$ processor-local Gibbs steps, effectively applying the block-diagonal update $B^{-1}C$ repeatedly using $Ax^{(t)} + h$ as a potential vector that includes out-of-date statistics from the other processors. The resulting update operator for one outer iteration of the Hogwild Gibbs sampling process is

$$x^{(t+1)} = (B^{-1}C)^q x^{(t)} + \sum_{j=0}^{q-1} (B^{-1}C)^j B^{-1} \left( Ax^{(t)} + h + v^{(t,j)} \right) \tag{7.5.1}$$

$$v^{(t,j)} \overset{\text{iid}}{\sim} \mathcal{N}(0, D) \tag{7.5.2}$$

where $D$ is the diagonal of $J$. Note that we shape the noise diagonally because in Hogwild Gibbs sampling we simply apply standard Gibbs updates in the inner iterations.

## ■ 7.6 Convergence and correctness of means

Because the Gaussian Hogwild Gibbs sampling iterates form a Gaussian linear dynamical system, the process is stable (i.e. its iterates converge in distribution) if and only

if [9, Lemma 7.3.6] the deterministic part of the update map (7.5.2) has spectral radius less than unity, i.e.

$$T \triangleq (B^{-1}C)^q + \sum_{j=0}^{q-1} (B^{-1}C)^j B^{-1}A \tag{7.6.1}$$

$$= (B^{-1}C)^q + (I - (B^{-1}C)^q)(I - (B^{-1}C))^{-1} B^{-1}A \tag{7.6.2}$$

$$= (B^{-1}C)^q + (I - (B^{-1}C)^q)(B - C)^{-1}A \tag{7.6.3}$$

$$= T_{\text{ind}}^q + (I - T_{\text{ind}}^q)T_{\text{bl}}, \tag{7.6.4}$$

where

$$T_{\text{ind}} \triangleq (B^{-1}C) \qquad T_{\text{bl}} \triangleq (B - C)^{-1}A, \tag{7.6.5}$$

satisfies $\rho(T) < 1$. The term $T_{\text{ind}}$ is the block Gauss-Seidel update when $A = 0$ and the processors' random variables are independent, while the term $T_{\text{bl}}$ is the block Jacobi update, which corresponds to solving the processor-local linear systems exactly at each outer iteration. The update (7.6.4) falls into the class of two-stage splitting methods [77, 35, 34], and the next proposition is equivalent to such two-stage solvers having the correct fixed point.

**Proposition 7.6.1.** *If a Gaussian Hogwild Gibbs process is stable, then its mean is* $\mu = J^{-1}h$.

*Proof.* If the process is stable the mean process has a unique fixed point, from (7.5.2) and (7.6.4) and using the definitions of $T_{\text{ind}}$ and $T_{\text{block}}$ we can write the fixed-point equation for the process mean $\mu_{\text{Hog}}$ as

$$(I - T)\mu_{\text{Hog}} = (I - T_{\text{ind}})(I - T_{\text{block}})\mu_{\text{Hog}} = (I - T_{\text{ind}})(B - C)^{-1}h, \tag{7.6.6}$$

hence $(I - (B - C)^{-1}A)\mu_{\text{Hog}} = (B - C)^{-1}h$ and $\mu_{\text{Hog}} = (B - C - A)^{-1}h = J^{-1}h$. □

The behavior of the spectral radius of the update map can be very complicated. In Figure 7.2, we compare $\rho(T)$ for $q = 1$ and $q = \infty$ for models generated from a simple random ensemble. Each point corresponds to a sampled model $J = QQ^{\mathsf{T}} + nrI$ with $Q_{ij} \overset{\text{iid}}{\sim} \mathcal{N}(0, 1)$ and $r \overset{\text{iid}}{\sim} \text{Uniform}[0.5, 1]$, and the value of each point's vertical coordinate is the spectral radius of the Hogwild update $T$ when $q = \infty$ (i.e. $T = T_{\text{block}}$) while the horizontal coordinate is the spectral radius of $T$ when $q = 1$. Hogwild Gibbs sampling on the model is convergent with $q = 1$ when the point is to the left of the vertical red line, and it is convergent as $q = \infty$ when the point is below the horizontal line. The figure shows that, while convergence in the two cases shows a positive correlation, Hogwild Gibbs can be convergent when $q = 1$ and not when $q = \infty$ and it can be

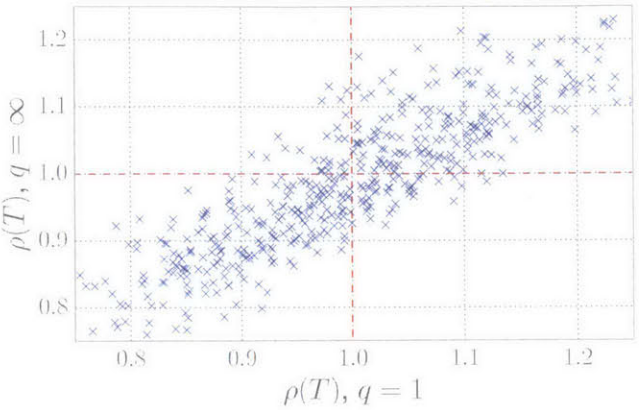



**Figure 7.2:** Comparing Hogwild stability on random models for extreme values of the inner iteration count $q$. Each point corresponds to a sampled model, where the horizontal coordinate is the spectral radius at $q = 1$ and the vertical coordinate is the spectral radius at $q = \infty$.

convergent when $q = \infty$ and not when $q = 1$. Therefore the behavior of the algorithm with varying $q$ is difficult to understand in general.

Despite the complexity of the update map's stability, in the next subsection we give a simple argument that identifies its convergence with the convergence of Gauss-Seidel iterates on a larger, non-symmetric linear system. Given that relationship we then prove a condition on the entries of $J$ that ensures the stability of the Gaussian Hogwild Gibbs sampling process.

### ■ 7.6.1 A lifting argument and sufficient condition

First observe that we can write multiple steps of Gauss-Seidel as a single step of Gauss-Seidel on a larger system: given $J = L - U$ where $L$ is lower-triangular (including the diagonal, unlike the notation of Section 7.3) and $U$ is strictly upper-triangular, consider applying Gauss-Seidel to a larger block $k \times k$ system:

$$\begin{pmatrix} L & & & -U \\ -U & L & & \\ & \ddots & \ddots & \\ & & -U & L \end{pmatrix} \xrightarrow{\text{G-S}} \begin{pmatrix} L^{-1} & & & \\ L^{-1}UL^{-1} & L^{-1} & & \\ \vdots & & \ddots & \\ (L^{-1}U)^{k-1}L^{-1} & \cdots & L^{-1}UL^{-1} & L^{-1} \end{pmatrix} \begin{pmatrix} & & & U \\ & & & \\ & & & \\ & & & \end{pmatrix} = \begin{pmatrix} & & & L^{-1}U \\ & & & \vdots \\ & & & (L^{-1}U)^k \end{pmatrix} \tag{7.6.7}$$

Therefore one step of Gauss-Seidel on the larger system corresponds to $k$ applications of the Gauss-Seidel update $L^{-1}U$ from the original system to the last block element of the lifted state vector.

Now we provide a lifted linear system on which Gauss-Seidel iterations correspond to applying Gaussian Hogwild Gibbs iterations to a block component.

**Proposition 7.6.2.** *Two applications of the Hogwild update $T$ of* (7.6.4) *are equivalent to the update to the last block element of the state vector in one Gauss-Seidel iteration on the $(2qn) \times (2qn)$ system*

$$\begin{pmatrix} E & -F \\ -F & E \end{pmatrix} \tilde{x} = \begin{pmatrix} h \\ \vdots \\ h \end{pmatrix} \text{ with } E = \begin{pmatrix} B & & \\ -C & B & \\ & \ddots & \ddots \\ & & -C \; B \end{pmatrix} \quad F = \begin{pmatrix} A+C & & \\ A & & \\ \vdots & & \\ A & & \end{pmatrix}. \tag{7.6.8}$$

*That is, if $P = \begin{pmatrix} 0 & \cdots & 0 & I \end{pmatrix}$ is $n \times 2qn$ with an $n \times n$ identity as its last block entry, then*

$$P \begin{pmatrix} E & \\ -F & E \end{pmatrix}^{-1} \begin{pmatrix} & F \end{pmatrix} P^{\mathsf{T}} = P \begin{pmatrix} & E^{-1}F \\ & (E^{-1}F)^2 \end{pmatrix} P^{\mathsf{T}} = T^2. \tag{7.6.9}$$

*Proof.* It suffices to consider $E^{-1}F$. Furthermore, since the claim concerns the last block entry, we need only consider the last block row of $E^{-1}F$. $E$ is block lower-bidiagonal and hence $E^{-1}$ has the same lower-triangular form as in (7.6.7),

$$E^{-1} = \begin{pmatrix} B^{-1} & & & \\ B^{-1}CB^{-1} & B^{-1} & & \\ \vdots & \ddots & \ddots & \\ (B^{-1}C)^{q-1}B^{-1} & \ldots & B^{-1}CB^{-1} & B^{-1} \end{pmatrix}, \tag{7.6.10}$$

and the product of the last block row of $E^{-1}$ with the last block column of $F$ yields

$$\begin{aligned} \begin{pmatrix} (B^{-1}C)^{q-1}B^{-1} & \ldots & (B^{-1}C)B^{-1} & B^{-1} \end{pmatrix} \cdot \begin{pmatrix} A+C & A & \ldots & A \end{pmatrix} & \qquad (7.6.11) \\ = (B^{-1}C)^q + \sum_{j=0}^{q-1} (B^{-1}C)^j B^{-1} A = T. & \qquad (7.6.12) \end{aligned}$$

□

**Proposition 7.6.3.** *Gaussian Hogwild Gibbs sampling is convergent if Gauss-Seidel converges on the system* (7.6.8).

To give a sufficient condition for the convergence of Gauss-Seidel on the lifted system and hence the Gaussian Hogwild Gibbs process, we first state a standard result for Gauss-Seidel and a simple corollary.

**Lemma 7.6.1** (Theorem 6.2 [22]). *If $J$ is strictly diagonally dominant then its Gauss-Seidel update matrix is a contraction in max norm; that is, if for every $i$ we have $|J_{ii}| > \sum_{j \neq i} |J_{ij}|$ then letting $J = L - U$ where $L$ is lower-triangular and $U$ is strictly*

*upper-triangular we have*

$$||L^{-1}U||_\infty < 1 \quad \textit{where} \quad ||A||_\infty \triangleq \sup_{x\neq 0} \frac{||Ax||_\infty}{||x||_\infty} = \max_i \sum_{j=1}^{n} |A_{ij}| \tag{7.6.13}$$

*and where* $||x||_\infty \triangleq \max_i |x_i|$.

Note that the Gauss-Seidel update being a contraction in any induced matrix norm immediately implies it is convergent since the spectral radius is upper bounded by any induced norm; that is, $\rho(A) \leq ||A||$ for any induced matrix norm $||\cdot||$ because if $v$ is the eigenvector corresponding to an eigenvalue of $A$ that achieves its spectral radius then

$$||A|| = \sup_{x\neq 0} \frac{||Ax||}{||x||} \geq \frac{||Av||}{||v||} = \rho(A). \tag{7.6.14}$$

We can extend the lemma slightly by considering generalized diagonally dominant matrices and adapting the max norm accordingly.

**Corollary 7.6.1.** *If $J$ is strictly generalized diagonally dominant then its Gauss-Seidel update matrix is a contraction in a weighted max norm; that is, if there exists an $r \in \mathbb{R}^n$ with $r > 0$ entrywise such that for every $i$ we have $r_i|J_{ii}| > \sum_{j\neq i} r_j|J_{ij}|$, then letting $J = L - U$ where $L$ is lower-triangular and $U$ is strictly upper-triangular we have*

$$||L^{-1}U||_\infty^r < 1 \quad \textit{where} \quad ||A||_\infty^r \triangleq \sup_{x\neq 0} \frac{||Ax||_\infty^r}{||x||_\infty^r} = \max_i \frac{1}{r_i} \sum_{j=1}^{n} |A_{ij}|r_j \tag{7.6.15}$$

*and where* $||x||_\infty^r \triangleq \max_i \frac{1}{r_i}|x_i|$.

*Proof.* Let $R \triangleq \text{diag}(r)$ and note that $JR$ is strictly diagonally dominant. Therefore by Lemma 7.6.1 we have that

$$1 > ||R^{-1}L^{-1}UR||_\infty = \max_{x\neq 0} \frac{||R^{-1}L^{-1}URx||_\infty}{||x||_\infty} \tag{7.6.16}$$

$$= \max_{x\neq 0} \frac{||L^{-1}URx||_\infty^r}{||x||_\infty} \tag{7.6.17}$$

$$= \max_{x\neq 0} \frac{||L^{-1}U||_\infty^r}{||x||_\infty^r} = ||L^{-1}U||_\infty^r \tag{7.6.18}$$

where we have used $||x||_\infty^r = ||R^{-1}x||_\infty$ and on the last line substituted $x \mapsto R^{-1}x$. □

With generalized diagonal dominance as a sufficient condition for Gauss-Seidel convergence, we can use the lifting construction of Proposition 7.6.2 to give a sufficient condition for the convergence of Gaussian Hogwild Gibbs.

**Theorem 7.6.1.** *If $J$ is strictly generalized diagonally dominant, that is if there exists an $r \in \mathbb{R}^n$ with $r > 0$ entrywise such that*

$$r_i|J_{ii}| > \sum_{j \neq i} r_j|J_{ij}|, \tag{7.6.19}$$

*then Gaussian Hogwild Gibbs sampling is convergent for any fixed variable partition and any fixed number of inner iterations. Further, we have $||T||_\infty^r < 1$.*

*Proof.* Since each scalar row of the coefficient matrix in (7.6.8) contains only entries from one row of $J$ and zeros, it is generalized diagonally dominant with a scaling vector that consists of $2q$ copies of $r$. Gauss-Seidel iterations on generalized diagonally dominant systems are convergent by Lemma 7.6.1 and so by Proposition 7.6.3 the corresponding Gaussian Hogwild Gibbs iterations are convergent.

To show the stronger result that the update is a contraction, first we define $\widetilde{T}$ to be the Gauss-Seidel update matrix for the system (7.6.8), i.e.

$$\widetilde{T} \triangleq \begin{pmatrix} E & \\ -F & E \end{pmatrix}^{-1} \begin{pmatrix} & F \\ & \end{pmatrix}, \tag{7.6.20}$$

and we define $\widetilde{r}$ to be $2q$ copies of $r$. For any $x \in \mathbb{R}^n$ we have

$$||x||_\infty^r = ||P^\mathsf{T} x||_\infty^{\widetilde{r}} > ||\widetilde{T}P^\mathsf{T} x||_\infty^{\widetilde{r}} \geq ||P\widetilde{T}P^\mathsf{T} x||_\infty^r = ||Tx||_\infty^r \tag{7.6.21}$$

where we have used the orthogonality of $P$ and Corollary 7.6.1. □

Note that the lifting construction in (7.6.8) immediately generalizes to the case where the number of inner iterations varies from processor to processor. Furthermore, the proof of Theorem 7.6.1 shows that $T$ is a contraction in $||\cdot||_\infty^r$ regardless of the partition or structure or inner iteration counts. Therefore we can immediately generalize the result to the non-stationary case, where the numbers of inner iterations and even the partition structure vary across outer iterations.

**Theorem 7.6.2.** *If $J$ is strictly generalized diagonally dominant, then for any inner iteration schedule $q$ with $1 \leq q(t,k) < q_{\max}$ (for $t = 1, 2, \ldots$, $k = 1, 2, \ldots, K$, and any $q_{\max} < \infty$) and any sequence of partitions $\mathcal{I}^{(t)} = \{\mathcal{I}_1^{(t)}, \ldots, \mathcal{I}_K^{(t)}\}$ Gaussian Hogwild Gibbs is convergent.*

*Proof.* We write $\mathcal{T}$ for the set of all possible update maps $T$, where $T^{(t)}$ is a function of both $q(t,k)$ and $\mathcal{I}^{(t)}$. The process is convergent if the joint spectral radius [95, 62] of $\mathcal{T}$ satisfies

$$\rho(\mathcal{T}) \triangleq \lim_{\ell \to \infty} \sup\{||T_1 \cdots T_\ell||^{\frac{1}{\ell}} : T_i \in \mathcal{T}\} < 1 \tag{7.6.22}$$

where $||\cdot||$ is any matrix norm. We use the matrix norm induced by the vector norm $||\cdot||_\infty^r$ defined in (7.6.15) and note that any induced norm is submultiplicative, so that for any matrices $T_1$ and $T_2$

$$||T_1 T_2||_\infty^r \leq ||T_1||_\infty^r ||T_2||_\infty^r. \tag{7.6.23}$$

Then, using the submultiplicative property and the contraction property from Theorem 7.6.1, for any $\ell$ and any $T_1, T_2, \ldots, T_\ell \in \mathcal{T}$ we have

$$(||T_1 \cdots T_\ell||_\infty^r)^{\frac{1}{\ell}} \leq (||T_1||_\infty^r \cdots ||T_\ell||_\infty^r)^{\frac{1}{\ell}} \leq 1 - \epsilon \tag{7.6.24}$$

for some $\epsilon > 0$ using the fact that $\mathcal{T}$ is finite. Therefore $\rho(\mathcal{T}) < 1$ and the process is convergent. □

Generalized diagonally dominant matrices are also known as H-matrices [9, Definition 5.13]; see Berman and Plemmons [9, Theorem 5.14] for a long list of equivalent characterizations. For an H-matrix to be a valid precision matrix it must also be positive semidefinite (PSD). Such matrices can also be described as having factor-width two [13]; that is, a PSD H-matrix $J$ can be factored as $J = GG^T$ where $G$ is a rectangular matrix in which each column has at most two nonzeros.

In terms of Gaussian graphical models, generalized diagonally dominant models include tree models and latent tree models (since H-matrices are closed under Schur complements), in which the density of the distribution can be written as a tree-structured set of pairwise potentials over the model variables and a set of latent variables. Latent tree models are useful in modeling data with hierarchical or multiscale relationships, and this connection to latent tree structure is evocative of many hierarchical Bayesian models. PSD H-matrices also include walk-summable matrices [75], for which the Gaussian Loopy Belief Propagation algorithm converges and yields correct mean estimates. More broadly, diagonally dominant systems are well-known for their tractability and applicability in many other settings [63], and Gaussian Hogwild Gibbs provides another example of their utility.

Because of the connection to linear system solvers known as two-stage multisplittings, these results can be identified with Theorem 2.3 of Frommer and Szyld [34], which shows that if the coefficient matrix is an H-matrix then the corresponding two-stage iterative solver is convergent. Indeed, by the connection between solvers and samplers

one can prove these convergence theorems as corollaries to Frommer and Szyld [34, Theorem 2.3] (or vice-versa), though our proof technique is much simpler. The other results on two-stage multisplittings [34, 77], including the results on asynchronous iterations, can also be applied immediately for results on the convergence of Gaussian Hogwild Gibbs sampling.

The sufficient conditions provided by Theorems 7.6.1 and 7.6.2 are coarse in that they provide convergence for any partition or update schedule. However, given the complexity of the processes, as exhibited in Figure 7.2, it is difficult to provide general conditions without taking into account some model structure.

### ■ 7.6.2 Exact local block samples

Convergence analysis simplifies greatly in the case where exact block samples are drawn at each processor because $q$ is sufficiently large or because another exact sampler [90, 29] is used on each processor. This regime of Hogwild Gibbs sampling is particularly interesting because it minimizes communication between processors.

In (7.5.2), we see that as $q \to \infty$ we have $T \to T_{\text{block}}$; that is, the deterministic part of the update becomes the block Jacobi update map, which admits a natural sufficient condition for convergence:

**Proposition 7.6.4.** *If* $((B-C)^{-\frac{1}{2}}A(B-C)^{-\frac{1}{2}})^2 \prec I$, *then block Gaussian Hogwild Gibbs sampling converges.*

*Proof.* Since similarity transformations preserve eigenvalues, with $\bar{A} \triangleq (B-C)^{-\frac{1}{2}}A(B-C)^{-\frac{1}{2}}$ we have $\rho(T_{\text{block}}) = \rho((B-C)^{\frac{1}{2}}(B-C)^{-1}A(B-C)^{-\frac{1}{2}}) = \rho(\bar{A})$ and since $\bar{A}$ is symmetric $\bar{A}^2 \prec I \Rightarrow \rho(\bar{A}) < 1 \Rightarrow \rho(T_{\text{block}}) < 1$. □

## ■ 7.7 Variances

Since we can analyze Gaussian Hogwild Gibbs sampling as a linear dynamical system, we can write an expression for the steady-state covariance $\Sigma_{\text{Hog}}$ of the process when it is stable. For a general stable LDS of the form $x^{(t+1)} = Tx^{(t)} + v^{(t)}$ with $v^{(t)} \sim \mathcal{N}(0, \Sigma_{\text{inj}})$ where $\Sigma_{\text{inj}}$ is the injected noise of the system, the steady-state covariance is given by the series $\sum_{t=0}^{\infty} T^t \Sigma_{\text{inj}} T^{t\mathsf{T}}$, which is the solution to the linear discrete-time Lyapunov equation $\Sigma = T\Sigma T^{\mathsf{T}} + \Sigma_{\text{inj}}$ in $\Sigma$ [20, 105].

The injected noise $\Sigma_{\text{inj}}$ for the the Hogwild iterations is determined by the inner iterations, which itself is a linear dynamical system with injected noise covariance $D$, the diagonal of $J$. For Hogwild sampling we have $\Sigma_{\text{inj}} = (I - T_{\text{ind}}^q)(B-C)^{-1}D(B-C)^{-1}(I - T_{\text{ind}}^q)^{\mathsf{T}}$. The target covariance is $J^{-1} = (B-C-A)^{-1}$.

Composing these expressions we see that the Hogwild covariance is complicated in general, but we can analyze some salient properties in at least two regimes of particular

**Algorithm 7.2** Hogwild Gibbs with Symmetric Local Sweeps

**Input:** Joint distribution over $x = (x_1, \ldots, x_n)$, partition $\{\mathcal{I}_1, \ldots, \mathcal{I}_K\}$ of $\{1, 2, \ldots, n\}$
Initialize $\bar{x}^{(1)}$
**for** $t = 1, 2, \ldots$ **do**
  **for** $k = 1, 2, \ldots, K$ in parallel **do**
    $\bar{x}_{\mathcal{I}_k}^{(t+1)} \leftarrow \text{LocalGibbs}(\bar{x}^{(t)}, \mathcal{I}_k, q(t,k))$

**function** LocalGibbs($\bar{x}$, $\mathcal{I}$, $q$)
  **for** $j = 1, 2, \ldots, q$ **do**
    **for** $i \in \mathcal{I}$ in order **do**
      $\bar{x}_i \leftarrow$ sample $x_i | x_{\neg i} = \bar{x}_{\neg i}$
    **for** $i \in \mathcal{I}$ in reverse order **do**
      $\bar{x}_i \leftarrow$ sample $x_i | x_{\neg i} = \bar{x}_{\neg i}$
  **return** $\bar{x}$

interest: first when $A$ is small so that higher-order powers of $A$ can be ignored, and second when local processors draw exact block samples (e.g. when $q \to \infty$).

### ■ 7.7.1 Low-order effects in $A$

Intuitively, the Hogwild strategy works best when cross-processor interactions are small, and so it is natural to analyze the case when $A$ is small and we can discard terms that include powers of $A$ beyond first or second order. To provide an analysis for the low-order regime, we first describe a variant of the Hogwild Gibbs algorithm that enables more detailed spectral analysis. We also fix notation for derivatives. The results in this subsection assume that the Hogwild process is convergent, which is guaranteed for small enough $A$ by continuity of the spectral radius.

For the remainder of Section 7.7.1 we analyze a slight variant of the Hogwild Gibbs algorithm in which processor-local Gibbs update sweeps are performed once in order and once in reverse order for each local iteration, as shown in Algorithm 7.2. This variant is more amenable to spectral analysis because its corresponding inner splitting has more structure than the Gauss-Seidel inner splitting of Algorithm 7.1. To see the difficulty with the Gauss-Seidel inner splitting, consider the splitting

$$\begin{pmatrix} 1 & 0.7 & \\ 0.7 & 1 & 0.7 \\ & 0.7 & 1 \end{pmatrix} \xrightarrow{\text{G-S}} \begin{pmatrix} 1 & & \\ 0.7 & 1 & \\ & 0.7 & 1 \end{pmatrix}^{-1} \begin{pmatrix} & 0.7 & \\ & & 0.7 \\ & & \end{pmatrix} = \begin{pmatrix} 0 & 0.7 & \\ & 0.7^2 & 0.7 \\ & 0.7^3 & 0.7^2 \end{pmatrix}. \quad (7.7.1)$$

This Gauss-Seidel update is not diagonalizable; its Jordan form is

$$\begin{pmatrix} 0 & 0.7 & \\ & 0.7^2 & 0.7 \\ & 0.7^3 & 0.7^2 \end{pmatrix} = \begin{pmatrix} -0.35 & \frac{5}{14} & -\frac{5}{14} \\ 0 & 0.5 & 0.5 \\ 0 & 0.35 & -0.35 \end{pmatrix} \begin{pmatrix} 0 & 1 & \\ & 0 & \\ & & 0.98 \end{pmatrix} \begin{pmatrix} -0.35 & \frac{5}{14} & -\frac{5}{14} \\ 0 & 0.5 & 0.5 \\ 0 & 0.35 & -0.35 \end{pmatrix}^{-1} \tag{7.7.2}$$

and so there is no basis of eigenvectors for the invariant subspace with eigenvalue 0. In general, a Gauss-Seidel update matrix may not be diagonalizable, and little can be said about its eigenvalues.

The inner splitting update matrix for Algorithm 7.2 is that of symmetric Gauss-Seidel, or Symmetric Successive Over-Relaxation (SSOR) with unit relaxation parameter [22]. This update has much clearer spectral properties, as we show in the following lemma. This lemma extends slightly a standard result that the eigenvalues of the SSOR update are real [22, p. 299].

**Lemma 7.7.1.** *Let $J \succ 0$ and let $J = D - L - L^T$, where $D$ is the diagonal of $J$ and $L$ and $L^T$ are its strictly lower- and upper-triangular parts, respectively. The symmetric Gauss-Seidel update matrix*

$$(D - L^T)^{-1} L (D - L)^{-1} L^T \tag{7.7.3}$$

*is diagonalizable, and furthermore its eigenvalues are real and in* $[0, 1)$.

*Proof.* We first show (7.7.3) is similar to a positive semidefinite matrix whenever $J$ is symmetric. By applying the similarity transformation $X \mapsto P^{-1} X P$ where $P \triangleq (D - L)^{-1} D^{\frac{1}{2}}$, we see (7.7.3) has the same eigenvalues as

$$D^{-\frac{1}{2}} L^T (D - L^T)^{-1} D^{\frac{1}{2}} D^{-\frac{1}{2}} L (D - L)^{-1} D^{\frac{1}{2}} = YZ \tag{7.7.4}$$

where $Y \triangleq D^{-\frac{1}{2}} L^T (D - L^T)^{-1} D^{\frac{1}{2}}$ and $Z \triangleq D^{-\frac{1}{2}} L (D - L)^{-1} D^{\frac{1}{2}}$. Note that

$$L^T (D - L^T)^{-1} = (D - (D - L^T))(D - L^T)^{-1} = D(D - L^T)^{-1} - I \tag{7.7.5}$$

and similarly $L(D - L)^{-1} = D(D - L)^{-1} - I$. Hence

$$Z = D^{-1/2} L (D - L)^{-1} D^{1/2} = D^{1/2} (D - L)^{-1} D^{1/2} - I \tag{7.7.6}$$

$$= \left[ D^{1/2} (D - L^T)^{-1} D^{1/2} - I \right]^T = \left[ D^{-1/2} L^T (D - L^T)^{\ -1} D^{1/2} \right]^T = Y^T \tag{7.7.7}$$

and so $YZ = YY^T$ is positive semidefinite and has nonnegative (real) eigenvalues. Furthermore, when $J \succ 0$ the eigenvalues have absolute value less than unity because symmetric Gauss-Seidel is convergent on positive definite systems. □

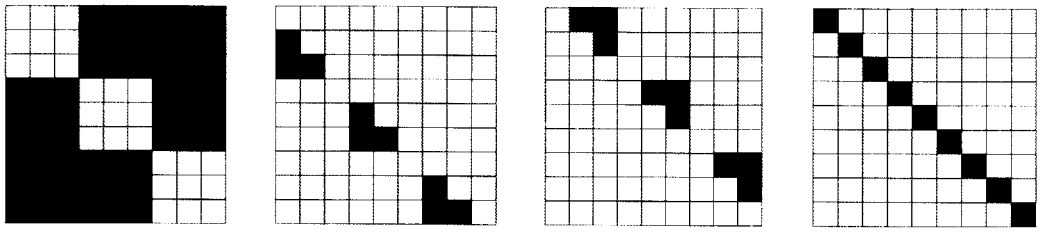



**Figure 7.3:** Support pattern (in black) of the splitting for Hogwild Gibbs with symmetric local sweeps, $J = D - C^\mathsf{T} - C - A$, with $n = 9$ and the processor partition $\{\{1,2,3\},\{4,5,6\},\{7,8,9\}\}$.

To model Algorithm 7.2 with its symmetric Gauss-Seidel inner splitting, given a precision matrix $J$ we split $J = D_{\text{bd}} - A$ into block diagonal and block off-diagonal parts as in Section 7.5, then further split $D_{\text{bd}} = D - C^\mathsf{T} - C$ into diagonal, strictly lower-triangular, and strictly upper-triangular parts. Note that $B = D - C^\mathsf{T}$, and so compared to the splitting presented in Section 7.5, we now split $J = D - C^\mathsf{T} - C - A$ instead of $J = B - C - A$. Additionally, though we have $B - C = D - C^\mathsf{T} - C$, in the equations in this section we continue to use $B - C$ for simplicity and consistency with other sections. See Figure 7.3 for an example of the sparsity pattern of $A$, $C^\mathsf{T}$, $C$, and $D$, and compare to Figure 7.1.

The inner-splitting update matrix for Algorithm 7.2 is then the block-diagonal matrix $S \triangleq (D-C)^{-1}C^\mathsf{T}(D-C^\mathsf{T})^{-1}C$. Comparing to (7.6.4), the deterministic part of the update map becomes

$$T \triangleq S^q + (I - S^q)T_{\text{bl}} \tag{7.7.8}$$

for the same definition of $T_{\text{bl}}$ as in (7.6.5), and the discrete-time Lyapunov equation for the Hogwild covariance remains

$$\Sigma_{\text{Hog}} = T\Sigma_{\text{Hog}}T^\mathsf{T} + \Sigma_{\text{inj}} \tag{7.7.9}$$

where now $\Sigma_{\text{inj}} = (I-S^q)(B-C)^{-1}D(B-C)^{-1}(I-S^q)^\mathsf{T}$. Similarly, in other expressions we can replace each occurrence of $T_{\text{ind}} = B^{-1}C$ with $S$. In the following analysis, we use the fact that $S$ has a complete basis of eigenvectors and that its eigenvalues are real and lie in $[0,1)$.

Next, we fix notation for derivatives, following the notation used in Pressley [91]. Let $\mathbb{R}^{n\times n}$ denote the space of real $n \times n$ matrices. For a function $f : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$, we write its derivative at $X \in \mathbb{R}^{n\times n}$ as the linear map $D_X f : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ defined by

$$D_X f(Y) \triangleq \frac{d}{dt} f(X + tY)\bigg|_{t=0} \qquad \forall Y \in \mathbb{R}^{n\times n} \tag{7.7.10}$$

where the differentiation is performed element-wise. Similarly, we write the second derivative at $X \in \mathbb{R}^{n\times n}$ as the symmetric bilinear form $D_X^2 f : \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ defined[3] by

$$D_X^2 f(Y,Y) = \frac{d^2}{dt^2} f(X+tY)\Big|_{t=0} \qquad \forall Y \in \mathbb{R}^{n\times n}. \tag{7.7.11}$$

Finally, we write the Taylor approximation for $f$ around the point 0 as

$$f(X) = f(0) + D_0 f(X) + \frac{1}{2} D_0^2 f(X,X) + \mathcal{O}(||X||^3) \tag{7.7.12}$$

where $||\cdot||$ is any submultiplicative matrix norm.

To analyze the Hogwild covariance error to low order in $A$, we write both the exact covariance and the Hogwild covariance as functions of the symmetric matrix $A$, respectively $\Sigma(A)$ and $\Sigma_{\text{Hog}}(A)$. We write the Taylor expansion of $\Sigma_{\text{Hog}}$ around 0 as

$$\Sigma_{\text{Hog}}(A) = \Sigma(0) + D_0\Sigma_{\text{Hog}}(A) + \frac{1}{2} D_0^2 \Sigma_{\text{Hog}}(A,A) + \mathcal{O}(||A||^3), \tag{7.7.13}$$

where $\Sigma(0) = (B-C)^{-1}$, and compare it to the exact series expansion for the target covariance $\Sigma = J^{-1}$ given by

$$\begin{aligned}
J^{-1} &= [B-C-A]^{-1} && (7.7.14)\\
&= (B-C)^{-\frac{1}{2}} \left[I - (B-C)^{-\frac{1}{2}} A (B-C)^{-\frac{1}{2}}\right]^{-1} (B-C)^{-\frac{1}{2}} && (7.7.15)\\
&= (B-C)^{-\frac{1}{2}} \Big[I + (B-C)^{-\frac{1}{2}} A (B-C)^{-\frac{1}{2}} \\
&\qquad + ((B-C)^{-\frac{1}{2}} A (B-C)^{-\frac{1}{2}})^2 + \cdots\Big] (B-C)^{-\frac{1}{2}} && (7.7.16)\\
&= \Sigma(0) + (B-C)^{-1} A (B-C)^{-1} \\
&\qquad + (B-C)^{-1} A (B-C)^{-1} A (B-C)^{-1} + \mathcal{O}(||A||^3). && (7.7.17)
\end{aligned}$$

In particular, to understand low-order effects in $A$, we compare the lowest-order terms that disagree in the two expansions.

We measure the total error as $||\Sigma_{\text{Hog}}(A) - \Sigma(A)||_{P,\text{Fro}}$, where

$$||X||_{P,\text{Fro}} \triangleq ||P^{-1} X P^{-\mathsf{T}}||_{\text{Fro}} \quad \text{and} \quad ||X||_{\text{Fro}} \triangleq \text{tr}(X^{\mathsf{T}} X) \tag{7.7.18}$$

and where $P \triangleq (D - C^{\mathsf{T}})^{-1} D^{\frac{1}{2}}$ is the similarity transformation used in the proof of

[3] A symmetric bilinear form $R$ on a vector space $V$ is defined by the quadratic form $Q$ with $Q(u) = R(u,u)$ for all $u \in V$ via the polarization identity $4R(u,v) = Q(u+v) - Q(u-v)$. Thus to define the second derivative it suffices to define the corresponding quadratic form, as in (7.7.11). In our analysis based on Taylor series expansion, we only use the quadratic form.

Lemma 7.7.1. In the following, we analyze the error on the block-diagonal and the error off the block-diagonal separately, decomposing

$$||\Sigma_{\text{Hog}}(A) - \Sigma(A)||_{P,\text{Fro}} = ||\Pi_{\text{bd}}(\Sigma_{\text{Hog}}(A) - \Sigma(A))||_{P,\text{Fro}} + ||\Pi_{\text{obd}}(\Sigma_{\text{Hog}}(A) - \Sigma(A))||_{P,\text{Fro}} \quad (7.7.19)$$

where $\Pi_{\text{bd}}$ and $\Pi_{\text{obd}}$ project to the block-diagonal and off-block-diagonal, respectively, and we have used the fact that $P$ is itself block-diagonal.

#### Block-diagonal error

To analyze the low-order effects of $A$ on the block-diagonal error, we first differentiate (7.7.9) to write an equation for $D_0\Sigma_{\text{Hog}}(A)$:

$$D_0\Sigma_{\text{Hog}}(A) - S^q D_0\Sigma_{\text{Hog}}(A)S^{q\mathsf{T}} = \widetilde{A}^{(1)} - S^q\widetilde{A}^{(1)}S^{q\mathsf{T}} - (I - S^q)\widetilde{A}^{(1)}(I - S^q)^{\mathsf{T}} \quad (7.7.20)$$

where $\widetilde{A}^{(1)} \triangleq (B-C)^{-1}A(B-C)^{-1} = D_0\Sigma(A)$ is the first-order term in the expansion for the exact covariance in (7.7.16). Note, however, that because $A$ is zero on its block-diagonal, $\Pi_{\text{bd}}(D_0\Sigma_{\text{Hog}}(A)) = 0 = \Pi_{\text{bd}}(\widetilde{A}^{(1)})$ so the first-order terms in both expansions, (7.7.16) and (7.7.13), are identical on the block-diagonal.

To compare second-order terms on the block-diagonal, we differentiate (7.7.9) twice to write an equation for $D_0^2\Sigma_{\text{Hog}}(A)$:

$$\Pi_{\text{bd}}\left(D_0^2\Sigma_{\text{Hog}}(A,A) - S^q D_0^2\Sigma_{\text{Hog}}(A,A)S^{q\mathsf{T}}\right) = 2\Pi_{\text{bd}}\left((I - S^q)\widetilde{A}^{(2)}(I - S^q)^{\mathsf{T}}\right) \quad (7.7.21)$$

where $\widetilde{A}^{(2)} \triangleq (B-C)^{-1}A(B-C)^{-1}A(B-C)^{-1} = \frac{1}{2}D_0^2\Sigma(A,A)$ is the second-order term in the expansion for the exact covariance in (7.7.16). Using (7.7.21) and the fact that $S$ has a complete set of eigenvectors, we can decompose the error in the second-order terms as

$$\left|\left|\Pi_{\text{bd}}\left(\frac{1}{2}D_0^2\Sigma_{\text{Hog}}(A,A) - \widetilde{A}^{(2)}\right)\right|\right|_{P,\text{Fro}}^2 = \sum_{k\in[K]}\sum_{(i,j)\in\mathcal{I}_k^2} |\widetilde{a}_{ij}^{(2)}|^2 f(\lambda_i^q, \lambda_j^q)^2 \quad (7.7.22)$$

where each $(\lambda_i, \lambda_j)$ is a pair of eigenvalues of a block of $S$, and $\widetilde{a}_{ij}^{(2)} \triangleq (Q^{\mathsf{T}}P^{-1}\widetilde{A}^{(2)}P^{-\mathsf{T}}Q)_{ij}$, where $Q$ is the orthogonal matrix such that $Q^{\mathsf{T}}P^{-1}SPQ$ is diagonal. The function $f : (-1,1)^2 \to \mathbb{R}_+$ is defined by

$$f(\lambda_i^q, \lambda_j^q) \triangleq \left|1 - \frac{(1-\lambda_i^q)(1-\lambda_j^q)}{1-\lambda_i^q\lambda_j^q}\right|. \quad (7.7.23)$$

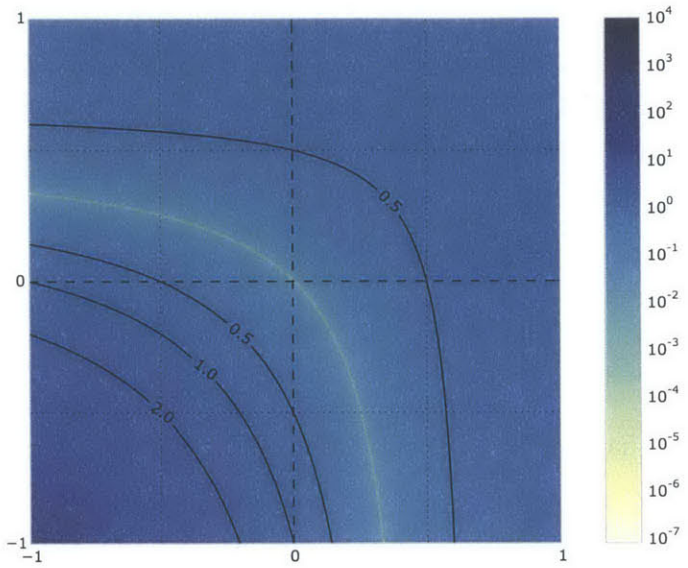



**Figure 7.4:** A plot of the function $f$ defined in (7.7.23).

Hence we can understand the error by analyzing the values of $f(\lambda_i^q, \lambda_j^q)$ for all the pairs of eigenvalues of $S$. For a detailed derivation, see Appendix A.

We plot the function $f$ in Figure 7.4. In the figure, the color corresponds to the value of $f$ on a logarithmic scale, and we label some level sets of $f$. Note in particular that $f(0,0) = 0$ and that $0 \leq f(x,y) < 1$ for $(x,y) \in [0,1)^2$. Due to Lemma 7.7.1, we need only consider the nonnegative quadrant $[0,1)^2$, which corresponds to the upper-right of the figure. Using these properties of $f$ and the decomposition (7.7.22) yields the following proposition.

**Proposition 7.7.1.** *Using* $||\cdot|| = ||\cdot||_{P,\mathrm{Fro}}$, *we have*

*(1) For all* $q \geq 1$, *the block-diagonal Hogwild covariance satisfies*

$$||\Pi_{\mathrm{bd}}\left(\Sigma_{\mathrm{Hog}}(A) - \Sigma(A)\right)|| < ||\Pi_{\mathrm{bd}}\left(\Sigma(0) - \Sigma(A)\right)|| + \mathcal{O}(||A||^3); \tag{7.7.24}$$

*(2) The dominant (second-order) error term decreases with increasing* $q$ *in the sense that*

$$\left|\left|\Pi_{\mathrm{bd}}\left(D_0^2\Sigma_{\mathrm{Hog}}(A,A) - D_0^2\Sigma(A,A)\right)\right|\right| \to 0 \tag{7.7.25}$$

*monotonically as* $q \to \infty$.

*Proof.* (1) is immediate from the decomposition (7.7.22), Lemma 7.7.1, and the observation that $0 \leq f(x,y) < 1$ for $(x,y) \in [0,1)^2$. To show (2), first we note that since $\lim_{q\to\infty} \lambda_i^q = 0$ for each eigenvalue $\lambda_i$ of $S$ and because $f$ is continuous at $(0,0)$, we have that $\lim_{q\to\infty} f(\lambda_i^q, \lambda_j^q) = f(0,0) = 0$ for every pair. Monotonicity follows from the fact that if for any $(x_0, y_0) \in [0,1)^2$ we define a path $\gamma(t) = (x_0^t, y_0^t)^\mathsf{T}$ for $t \in \mathbb{R}_+$, then we have

$$\frac{d}{dt}\gamma(t) = \begin{pmatrix} \ln(x_0)x_0^t & \ln(y_0)y_0^t \end{pmatrix}^\mathsf{T} < 0 \tag{7.7.26}$$

element-wise, and since

$$\nabla f(x,y) = \begin{pmatrix} \frac{(1-y)^2}{(1-xy)^2} & \frac{(1-x)^2}{(1-xy)^2} \end{pmatrix}^\mathsf{T} > 0 \tag{7.7.27}$$

element-wise, we have $\frac{d}{dt} f(\gamma(t)) = \langle \nabla f(x_0^t, y_0^t), \frac{d}{dt}\gamma(t) \rangle < 0$ and $f$ is monotonically decreasing along $\gamma$. □

Proposition 7.7.1 shows that, to second order in $A$, the block-diagonal of the Hogwild covariance is always improved relative to simply ignoring cross-processor effects by approximating $A = 0$, and that the amount of second-order improvement is monotonically increasing with the number of local iterations $q$.

#### Off-block-diagonal error

Returning to the first-derivative equation (7.7.20), we see that both $D_0\Sigma_{\text{Hog}}(A)$ and $\widetilde{A}^{(1)}$ are nonzero off the block-diagonal, and therefore to analyze the off-block-diagonal covariance error for small $A$ we compare the first-order terms. Analogous to the argument in the previous section, as derived in Appendix A we can decompose the error in the first-order terms as

$$\left|\left|\Pi_{\text{obd}}\left(D_0\Sigma_{\text{Hog}} - \widetilde{A}^{(1)}\right)\right|\right|_{P,\text{Fro}}^2 = \sum_{\substack{k_1,k_2 \in [K] \\ k_1 \neq k_2}} \sum_{i \in \mathcal{I}_{k_1}} \sum_{j \in \mathcal{I}_{k_2}} |\widetilde{a}_{ij}^{(1)}|^2 g(\lambda_i^q, \lambda_j^q)^2 \tag{7.7.28}$$

where each $(\lambda_i, \lambda_j)$ is a pair of eigenvalues from distinct blocks of $S$ and we set $\widetilde{a}_{ij}^{(1)} \triangleq (Q^\mathsf{T} P^{-1} \widetilde{A}^{(1)} P^{-\mathsf{T}} Q)_{ij}$, where $Q$ is the orthogonal matrix such that $Q^\mathsf{T} P^{-1} S P Q$ is diagonal. The function $g : (-1,1)^2 \to \mathbb{R}_+$ is defined by

$$g(\lambda_i^q, \lambda_j^q) \triangleq \left| \frac{(1-\lambda_i^q)(1-\lambda_j^q)}{1-\lambda_i^q\lambda_j^q} \right|. \tag{7.7.29}$$

We plot the function $g$ in Figure 7.5. In the figure, the color corresponds to the value of $g$ on a logarithmic scale, and we label some level sets of $g$. Note in particular

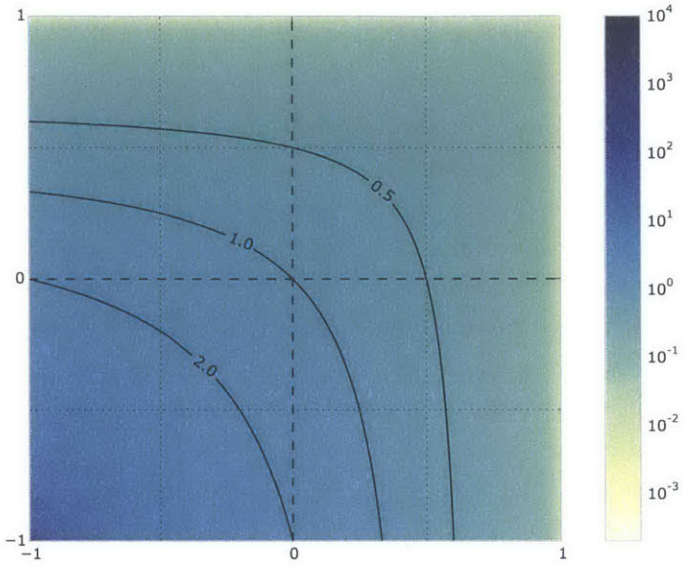



**Figure 7.5:** A plot of the function $g$ defined in (7.7.29).

that $g(0,0) = 1$ and that $0 \leq g(x,y) < 1$ for $(x,y) \in [0,1)^2$. Using these properties of $g$ and the decomposition (7.7.28) yields the following proposition.

**Proposition 7.7.2.** *Using* $||\cdot|| = ||\cdot||_{P,\mathrm{Fro}}$, *we have*

*(1) For all* $q \geq 1$, *the off-block-diagonal diagonal Hogwild covariance satisfies*

$$||\Pi_{\mathrm{obd}}(\Sigma_{\mathrm{Hog}}(A) - \Sigma(A))|| \leq ||\Pi_{\mathrm{obd}}(\Sigma(0) - \Sigma(A))|| + \mathcal{O}(||A||^2) \tag{7.7.30}$$

*where the inequality is strict if* $S$ *has a nonzero eigenvalue;*

*(2) The dominant (first-order) error term increases with increasing* $q$ *in the sense that*

$$||\Pi_{\mathrm{obd}}(D_0\Sigma_{\mathrm{Hog}}(A,A) - D_0\Sigma(A,A))|| \to ||\Pi_{\mathrm{obd}}(\Sigma(0) - \Sigma(A))|| \tag{7.7.31}$$

*monotonically as* $q \to \infty$.

*Proof.* As in the proof for Proposition 7.7.1, (1) follows immediately from the decomposition (7.7.28), Lemma 7.7.1, and the observation that $0 \leq g(x,y) < 1$ for $(x,y) \in [0,1)^2$. To show (2), note that $\lim_{q\to\infty} g(\lambda_i^q, \lambda_j^q) = 1$. By comparing the level sets of $g$ to those

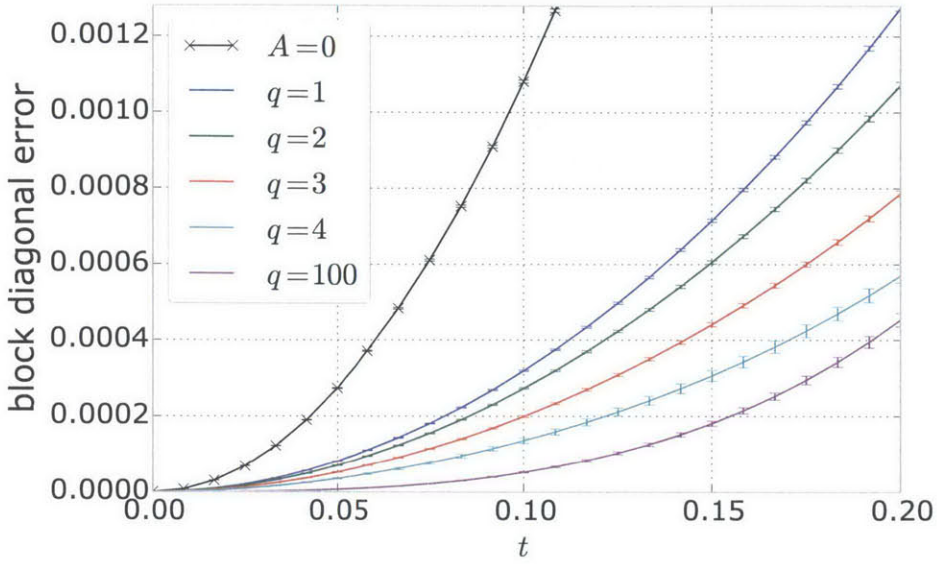
of $f$, we see that if for any $(x_0, y_0) \in [0,1)^2$ we define $\gamma(t) = (x_0^t, y_0^t)^\mathsf{T}$ for $t \in \mathbb{R}_+$, then we have $\frac{d}{dt} g(\gamma(t)) = \langle \nabla g(x_0^t, y_0^t), \frac{d}{dt}\gamma(t) \rangle > 0$ and so $g$ is monotonically increasing along the path $\gamma$. □

Proposition 7.7.2 shows that, to first order in $A$, the off-block-diagonal of the Hogwild covariance is always an improvement relative to the $A = 0$ approximation (assuming $S$ has a nonzero eigenvalue), yet the amount of first-order improvement is monotonically decreasing with the number of local iterations $q$. Therefore there is a tradeoff in covariance performance when choosing $q$, where larger values of $q$ improve the Hogwild covariance on the block diagonal but make worse the covariance error off the block diagonal, at least to low order in $A$.

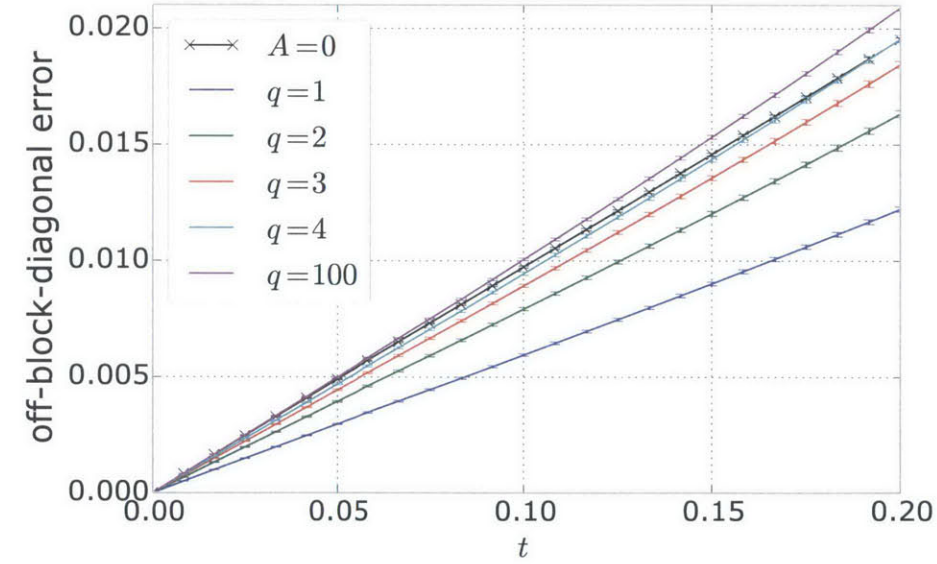
We validate these qualitative findings in Figure 7.6. Model families parameterized by $t$ are generated by first sampling $J = B - C - A = QQ^\mathsf{T}$ with $Q_{ij} \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$ where $Q$ is $n \times n$ and then letting $J(t) = B - C - tA$, so that $t = 0$ corresponds to a model with zero off-block-diagonal entries and off-block-diagonal effects increase with $t$. The sampling procedure is repeated 10 times with $n = 150$ and a partition with $K = 3$ and each $|\mathcal{I}_k| = 50$. The plotted lines show the average error (with standard deviation error bars) between the block diagonal of the true covariance $\Sigma(t) = J(t)^{-1}$ and the block diagonal of the Hogwild covariance $\Sigma_{\text{Hog}}(t)$ as a function of $t$ for $q = 1, 2, 3, 4, 100$, where varying $q$ shows the effect of local mixing rates. That is, in Figure 7.6(a) each line plots the block-diagonal error $||\Pi_{\text{bd}}(\Sigma(t) - \Sigma_{\text{Hog}}(t))||_{P,\text{Fro}}$ and in Figure 7.6(b) each line plots the off-block-diagonal error $||\Pi_{\text{obd}}(\Sigma(t) - \Sigma_{\text{Hog}}(t))||_{P,\text{Fro}}$. Note that separate black line is plotted for $||\Pi_{\text{bd}}(\Sigma(t) - \Sigma(0))||_{P,\text{Fro}}$ and $||\Pi_{\text{obd}}(\Sigma(t) - \Sigma(0))||_{P,\text{Fro}}$, respectively; that is, the black lines plot the respective errors when ignoring cross-processor effects and approximating $A = 0$.

Figure 7.6(a) shows that to first order in $t$ the block diagonal of the process covariance $\Sigma_{\text{Hog}}$ is identical to the true covariance $\Sigma$, since all slopes are zero at $t = 0$. Second-order effects contribute to improve the Hogwild covariance relative to the $A = 0$ approximation. Furthermore, we see that the second-order effects result in lower errors on the block diagonal when there is more processor-local mixing, i.e. larger values of $q$. Similarly, Figure 7.6(b) shows that first-order effects contribute to improve the Hogwild off-block-diagonal covariance relative to the $A = 0$ approximation. The Hogwild slopes at $t = 0$ are lower than that of the $A = 0$ approximation, and the relative improvement decreases monotonically as $q$ grows and the slopes approach that of the $A = 0$ approximation. These features and their dependence on $q$ are described in general by Propositions 7.7.1 and 7.7.2.

Figure 7.6(b) also shows that, for larger values of $t$, higher-order terms contribute to make the Hogwild off-block-diagonal covariance error larger than that of the $A = 0$ approximation, especially for larger $q$. The setting where $q$ is large and global commu-




(a) $\Pi_{\text{bd}}$ projects to the block diagonal




(b) $\Pi_{\text{obd}}$ projects to the off-block-diagonal

**Figure 7.6:** Typical plots of the projected error $||\Pi(\Sigma(t) - \Sigma_{\text{Hog}}(t))||_{P,\text{Fro}}$ for random model families of the form $J(t) = B - C - tA$. In (a) $\Pi$ projects to the block diagonal; in (b) $\Pi$ projects to the off-block-diagonal. The sampled models had $\rho(S) \approx 0.67$. Hogwild covariances were computed numerically by solving the associated discrete time Lyapunov equation [111, 112].

nication is infrequent is of particular interest because it reflects greater parallelism (or an application of more powerful local samplers [90, 29]). In the next subsection we show that this case admits a special analysis and even an inexpensive correction to recover asymptotically unbiased estimates for the full covariance matrix.

### ■ 7.7.2 Exact local block samples

As local mixing increases, e.g. as $q \to \infty$ or if we use an exact block local sampler between global synchronizations, we are effectively sampling in the block lifted model of Eq. (7.3.13) and therefore we can use the lifting construction to analyze the error in variances.

**Proposition 7.7.3.** *When local block samples are exact, the Hogwild covariance* $\Sigma_{\text{Hog}}$ *satisfies*

$$\Sigma = (I + (B-C)^{-1}A)\Sigma_{\text{Hog}} \quad \textit{and} \quad ||\Sigma - \Sigma_{\text{Hog}}|| \leq ||(B-C)^{-1}A||\ ||\Sigma_{\text{Hog}}|| \tag{7.7.32}$$

*where* $\Sigma = J^{-1}$ *is the exact target covariance and* $||\cdot||$ *is any submultiplicative matrix norm. In particular, we may compute*

$$\Sigma = \Sigma_{\text{Hog}} + (B-C)^{-1}A\Sigma_{\text{Hog}} \tag{7.7.33}$$

*as a correction which requires only a large matrix multiplication and solving the processor-local linear systems because* $B - C$ *is block-diagonal.*

*Proof.* Using the block lifting in (7.3.13), the Hogwild process steady-state covariance is the marginal covariance of half of the lifted state vector, so using Schur complements we can write

$$\begin{aligned}
\Sigma_{\text{Hog}} &= ((B-C) - A(B-C)^{-1}A)^{-1} && (7.7.34)\\
&= (B-C)^{-\frac{1}{2}}\left[I + ((B-C)^{-\frac{1}{2}}A(B-C)^{-\frac{1}{2}})^2\right.\\
&\qquad \left. +((B-C)^{-\frac{1}{2}}A(B-C)^{-\frac{1}{2}})^4 + \cdots\right](B-C)^{-\frac{1}{2}}. && (7.7.35)
\end{aligned}$$

We can compare this series to the exact expansion in (7.7.16) to see that $\Sigma_{\text{Hog}}$ includes exactly the even powers, so therefore

$$\begin{aligned}
\Sigma - \Sigma_{\text{Hog}} &= (B-C)^{-\frac{1}{2}}\left[((B-C)^{-\frac{1}{2}}A(B-C)^{-\frac{1}{2}})\right.\\
&\qquad \left. +((B-C)^{-\frac{1}{2}}A(B-C)^{-\frac{1}{2}})^3 + \cdots\right](B-C)^{-\frac{1}{2}} && (7.7.36)\\
&= (B-C)^{-1}A\Sigma_{\text{Hog}}. && (7.7.37)
\end{aligned}$$

□

Note that this result does not place any assumptions on the off-block-diagonal $A$.

## ■ 7.8 Summary

We have introduced a framework for understanding Gaussian Hogwild Gibbs sampling and shown some results on the stability and errors of the algorithm, including (1) quantitative descriptions for when a Gaussian model is not too dependent to cause Hogwild sampling to be unstable (Proposition 7.6.2, Theorems 7.6.1 and 7.6.2, Proposition 7.6.4); (2) given stability, the asymptotic Hogwild mean is always correct (Proposition 7.6.1); (3) in the low-order regime with small cross-processor interactions, there is a tradeoff between the block-diagonal and off-block-diagonal Hogwild covariance errors (Propositions 7.7.1 and 7.7.2); and (4) when local samplers are run to convergence we can bound the error in the Hogwild variances and even efficiently correct estimates of the full covariance (Proposition 7.7.3). We hope these ideas may be extended to provide further insight into Hogwild Gibbs sampling, in the Gaussian case and beyond.

Chapter 8

# Conclusions and Future Directions

This thesis presents two lines of work addressing some scaling challenges in Bayesian inference. In the first we focus on inference algorithms for Bayesian time series models based on the HMM and HSMM. We demonstrate that efficient inference in such models goes hand in hand with model structure. In particular, in Chapter 3 we develop a Bayesian nonparametric model that includes explicit duration modeling, and in Chapter 4 we show that for duration modeling in long observation sequences it is crucial to have efficient representations so that message passing inference remains tractable. In Chapters 5 we exploit message passing and efficient HSMM representations to build efficient scalable inference algorithms in the SVI framework, which enables these Bayesian models to be fit to very large datasets, and in Chapter 6 we demonstrate how these models and algorithms can be composed to model extremely rich dynamical behavior.

In Chapter 7 we turn to a more general setting and develop a theoretical analysis of Hogwild Gibbs sampling in Gaussian models. This highly-parallelizable strategy for generating approximate samples using only Gibbs sampling updates has proven to be very effective in some cases, and by a theoretical analysis of the Gaussian case we aim to provide understanding of its performance and tradeoffs.

Here we summarize our main contributions and offer some thoughts on possible future directions of research.

## ■ 8.1 Summary of main contributions

In this section we provide an overview of the main contributions in each chapter.

### Chapter 3: The Hierarchical Dirichlet Process Hidden semi-Markov Model

In Chapter 3 we define the HDP-HSMM, which is a generalization of the HDP-HMM to allow arbitrary state-specific duration distribution modeling. We develop both collapsed and weak limit Gibbs samplers as well as an auxiliary variable scheme that makes the sampling updates both simple and efficient. In addition, we show a composition of HDP-HSMMs into a factorial structure, and apply this factorial model to an energy

signal disaggregation task.

### Chapter 4: Faster HSMM Inference with Efficient Representations

In Chapter 4 we develop a framework for representations of HSMMs that allow for efficient message passing and inference. By defining HMM embeddings of HSMMs we refine and extend previous ideas on embedded state HMMs (ESHMMs) and provide a general algorithm for computing HSMM messages that is both time- and memory-efficient for duration distributions that admit efficient representations. We then generalize further to a notion of LTI realizations of HSMMs, and by building connections with LTI systems realization and positive realization theory, we show both the generality of the LTI realization perspective as well as the ultimate limitations of such approaches. These connections may also yield new approximation schemes for HSMM inference.

### Chapter 5: SVI for HMMs, HSMMs, and Nonparametric Extensions

In Chapter 5 we use the stochastic variational inference (SVI) framework to develop scalable mean field variational inference algorithms for Bayesian HMMs, HSMMs, and their nonparametric extensions, the HDP-HMM and HDP-HSMM. Building on the ideas from Chapter 4, we also propose a more efficient approximate update for HSMMs with durations that are modeled as negative binomials (or negative binomial mixtures).

### Chapter 6: Scalable Inference in Models with Multiple Timescales

In Chapter 6 we synthesize the ideas in earlier chapters to define a new Bayesian nonparametric segment model with efficient and scalable inference algorithms. The model is based on the HDP-HSMM, but instead of observations within each segment being generated independently, observation segments are generated from HDP-HMMs. While many similar models have been proposed, the main advantage to this new model is that it admits efficient inference algorithms based on the ideas we develop in this thesis. In particular, building on ideas from Chapters 3 and 4, we construct efficient message passing and Gibbs sampling algorithms, and building on ideas from Chapter 5 we develop SVI updates that allow the model to be fit to large datasets.

### Chapter 7: Analyzing Hogwild Parallel Gaussian Gibbs Sampling

In Chapter 7 we define the Hogwild Gibbs sampling algorithm based on a simple but poorly-understood parallelization scheme for Gibbs samplers. We analyze this scheme in the case of sampling from Gaussian distributions, and drawing on linear algebraic tools we provide several results on its convergence and correctness. In particular, in Theorems 7.6.1 and 7.6.2 and Proposition 7.6.1 we give sufficient conditions for the stability of the stochastic process and the correctness of the process mean. To understand

the errors introduced in covariances, in Propositions 7.7.1 and 7.7.2 we give an analysis of the error in the process covariance when the model's cross-processor interactions are small, and in particular we show how the frequency of communication produces a trade-off in the errors on and off the block diagonal. Finally, for the asymptotic case where the frequency of cross-processor communication is minimized and parallel computation is maximized, in Propositions 7.6.4 and 7.7.3 we give a more precise convergence condition, an error bound on covariances, and even a correction so that the exact model covariance can be computed efficiently from the process covariance.

## ■ 8.2 Directions of future research

Here we suggest some lines of research that build on the ideas presented in this thesis.

### Extending the HDP-HMM and HDP-HSMM to continuous-time

The flexible models and algorithms in this thesis include a wide variety of specific models with particular observation distributions or hierarchically tied parameters. These models and algorithms also accommodate standard extensions to include switching linear dynamical systems as well as time-inhomogeneous models, in which the transition matrix or other parameters may vary over time. However, these models are all discrete-time models. Continuous-time versions of these models would be useful for settings where the data sequence does not represent observations sampled at uniformly-spaced intervals of time, i.e. where observations instead each have timestamps.

While there is some work on constructing continuous-time Bayesian nonparametric time series models [99], the HDP-HSMM offers a particular perspective on the construction of such models which may allow for the transfer of algorithms from the discrete-time setting to the continuous-time setting. Continuous-time Markov chains are typically parameterized in terms of a continuous-time arrival process, which generates transition times, and a transition matrix which excludes self-transitions [85]. The HDP-HSMM construction of Chapter 3 similarly separates the HDP prior over the transition matrix from the description of transition times, which are controlled by the discrete-time duration distributions. Therefore it may be possible to construct a corresponding continuous-time process, and corresponding Bayesian nonparametric prior, using the same diagonal-truncated HDP construction as in Chapter 3 but exchanging the discrete-time duration distributions for continuous-time ones. The advantage to such a construction, in addition to its simplicity, is that many of the algorithms we develop for the HDP-HSMM may be immediately applicable to the continuous-time model. In particular, the auxiliary variable sampler for performing Gibbs sampling with the truncated HDP prior developed in Chapter 3, the efficient duration representations explored in Chapter 4, and the SVI methods derived in Chapter 5 could be adapted.

### Direct inference in efficient HSMM representations

In Chapter 4 we develop a framework for efficient representations of duration models in HSMMs, particularly both HMM embeddings and LTI realizations. For particular duration models, namely negative binomials or mixtures of negative binomials, these representations led to much more efficient HSMM message passing and hence more efficient inference algorithms. While mixtures of negative binomials can represent a wide variety of potentially multi-modal duration distributions, it is natural to look for other duration models that have efficient representations yet are convenient for Bayesian inference over their parameters.

Negative binomial duration models (and their mixtures) are particularly convenient for Bayesian inference over their parameters because, due to their regular structure, we can develop simple priors and Gibbs sampling updates for their parameters, as we show in Section 4.4.2. However, it is not necessary to retain such simple parameter updates, especially if there are potential gains for duration model expressiveness and efficiency.

One approach is to parameterize the efficient representations directly, either as HMM embeddings or as LTI realizations. That is, in the case of HMM embeddings, one could directly parameterize the pseudostate transition matrices $\bar{A}^{(i)}$, the entrance probabilities $\bar{c}^{(i)}$, and the exit probabilities $\bar{b}^{(i)}$. As duration models, this class of distributions can be very rich, and the complexity of message passing inference using such representations is clear. Furthermore, one could perform Bayesian inference over the parameters using generic algorithms such as Metropolis-Hastings or Hamiltonian Monte Carlo [11]. Alternatively, one could exploit the HMM embeddings' probabilistic interpretation to develop an auxiliary variable sampler, in which, conditioned on a set of durations interpreted as dwell times, one could alternate between resampling the pseudostate sequences that gave rise to those dwell times and the pseudostate transition matrix given the complete pseudostate sequences. However, it remains unclear whether such duration distributions can offer significant modeling advantages over mixtures of negative binomial distributions.

### Compositional time series model learning

Recent work has explored automatic model generation and selection in compositional spaces [46, 24, 45]. In this line of work, a compositional model space may be specified by a set of grammar rules, each of which can refine a model component to express more detailed structure. More concretely, in Grosse [45, Chapter 3] models are expressed as matrix decompositions while grammar rules correspond to further decompositions of component factors. For example [45, Section 3.4.1], one might express a structureless model for a data matrix as $G$, where the symbol $G$ denotes that the matrix entries are generated independently and identically as Gaussians. By applying the production

rule $G \to MG + G$, where the symbol $M$ denotes a "multinomial" matrix with one nonzero entry per row, one can express clustering structure. Alternatively, by applying the production rule $G \to GG + G$, one can express low-rank structure. Grosse [45] shows that, by recursively applying these and other rules and using Bayesian model selection criteria and a greedy search, one can automatically discover very rich models from data.

These automatic model discovery and refinement techniques may prove a powerful tool for exploratory data analysis, especially when developing models in the unsupervised setting. The Bayesian models and algorithms for unsupervised time series analysis developed in this thesis may also fit naturally into such a framework. Models with the potential for dynamics at multiple timescales, such as those described in Chapter 6, may be particularly relevant, since an automatic search procedure might identify how detailed a dynamical model should be and what form those dynamics might take, including discrete Markov state dynamics or latent linear dynamics.

### More general stochastic variational inference

The SVI algorithms developed in Chapter 5 are applicable to many Bayesian time series models, including HMMs, HSMMs, and their nonparametric extensions paired with any observation or duration models. However, the unbiased stochastic natural gradient update relies on the assumption that the minibatches used to compute each update are conditionally independent given the global parameters. This assumption would be violated if the overall graphical model were to have edges connecting the latent variables of distinct minibatches.

Therefore to apply SVI to such models requires either an extension of the algorithm or a refinement of the analysis to show that, at least for some models or in some regimes, the existence of such cross-minibatch edges does not affect the convergence guarantees of the SVI algorithm. A refined analysis is plausible because stochastic gradient algorithms are known to converge under very weak assumptions [10]. Stochastic gradient algorithms are also known to converge even when the computation is aggressively parallelized or distributed among many infrequently-communicating processors [10, 84], and it would be beneficial to study these computational regimes, both theoretically and empirically, for SVI in Bayesian models.

### Understanding Hogwild Gibbs for Non-Gaussian Models

While the theory we develop in Chapter 7 only describes sampling in Gaussian models, it may be possible to translate some of our results and techniques to non-Gaussian settings.

In particular, several lines of work on developing scalable sampling algorithms rely

on appeals to the Bayesian central limit theorem [116, 1, 81], in which, as the amount of data grows large while the number of parameters stays fixed, posterior distributions for sufficiently differentiable models become Gaussian. Our Gaussian theory immediately applies to these settings. However, this framing is limited because it does not include most nonparametric latent variable models in which the relevant latent dimension grows with the amount of data. Settings in which the amount of data is sufficiently large to completely overwhelm any prior also obscure the utility of taking a Bayesian analysis approach. However, it may still be of interest to employ Hogwild Gibbs or similar samplers to quantify uncertainty in these regimes, and so it would be of interest to extend our Hogwild Gibbs sampling theory to these cases and evaluate the performance of such sampling methods empirically.

There are other avenues for extending Hogwild Gibbs analysis. In particular, it may be possible to build on ideas from parallel and distributed nonlinear optimization [10] to show convergence results for such samplers. A linearized analysis of covariance effects, such as the one we developed for Gaussians in Section 7.7.1, may also be possible in such settings.

Finally, it may be possible to develop an understanding of when Hogwild Gibbs procedures may be corrected to produce exact samples, such as with the inclusion of some Metropolis steps or other adjustments [94, 117].

Appendix A

# Hogwild Gibbs Covariance Spectral Analysis Details

Here we derive the expansions used in Eqs. (7.7.22) and (7.7.28). To unify the notation, we drop the superscripts on $\widetilde{A}^{(1)}$ and $\widetilde{A}^{(2)}$ and write simply $\widetilde{A}$ since the argument is identical in the two cases.

Recall from Lemma 7.7.1 that $S$ can be diagonalized via

$$\Lambda = \widetilde{P}^{-1} S \widetilde{P} \quad \text{where} \quad \widetilde{P}^{-1} = Q^{\mathsf{T}} P^{-1} \tag{A.1}$$

in which $Q$ is orthogonal and $P$ is the matrix $P \triangleq B^{-1} D^{\frac{1}{2}}$. The columns of $\widetilde{P}$ form a basis of $\mathbb{R}^n$, and

$$\mathcal{S} \triangleq \{\widetilde{p}_i \widetilde{p}_j^{\mathsf{T}} : \widetilde{p}_i, \widetilde{p}_j \text{ are the } i\text{th and } j\text{th columns of } \widetilde{P}\} \tag{A.2}$$

is a basis for $\mathbb{R}^{n \times n}$. The maps $X \mapsto SXS^{\mathsf{T}}$ and $X \mapsto (I - S)X(I - S)^{\mathsf{T}}$ are diagonal in the basis $\mathcal{S}$, since if $\lambda_i$ is the eigenvalue of $S$ corresponding to eigenvector $\widetilde{p}_i$ then

$$S \widetilde{p}_i \widetilde{p}_j^{\mathsf{T}} S^{\mathsf{T}} = \lambda_i \lambda_j \widetilde{p}_i \widetilde{p}_j^{\mathsf{T}} \tag{A.3}$$

$$(I - S) \widetilde{p}_i \widetilde{p}_j^{\mathsf{T}} (I - S)^{\mathsf{T}} = (1 - \lambda_i)(1 - \lambda_j) \widetilde{p}_i \widetilde{p}_j^{\mathsf{T}}. \tag{A.4}$$

The coefficients $\widetilde{a}_{ij}$ are then the coordinates of $\widetilde{A}$ in the basis $\mathcal{S}$. That is,

$$\widetilde{A} = \sum_{i,j} \widetilde{a}_{ij} \widetilde{p}_i \widetilde{p}_j^{\mathsf{T}} \quad \text{where} \quad \widetilde{a}_{ij} = (\widetilde{P}^{-1} \widetilde{A} \widetilde{P}^{-\mathsf{T}})_{ij}. \tag{A.5}$$

The formula follows from the computation

$$\widetilde{P}^{-1}\widetilde{A}\widetilde{P}^{-\mathsf{T}} = \sum_{i,j} \widetilde{a}_{ij}\widetilde{P}^{-1}\widetilde{p}_i\widetilde{p}_j\widetilde{P}^{-\mathsf{T}} \tag{A.6}$$

$$= \sum_{i,j} \widetilde{a}_{ij}e_ie_j^{\mathsf{T}} \tag{A.7}$$

where $e_i$ is the $i$th standard basis vector with 1 in its $i$th coordinate and 0 elsewhere.

With this setup, Eqs. (7.7.22) and (7.7.28) correspond to expanding Eqs. (7.7.21) and (7.7.20) in the $\mathcal{S}$ basis. Finally, note that

$$||\widetilde{A}||_{P,\text{Fro}} \triangleq ||P^{-1}\widetilde{A}P^{-\mathsf{T}}||_{\text{Fro}} = ||Q^{\mathsf{T}}P^{-1}\widetilde{A}P^{-\mathsf{T}}Q||_{\text{Fro}} = \sqrt{\sum_{i,j}\widetilde{a}_{ij}^2}. \tag{A.8}$$

Appendix B

# Numerical Evaluation of HDP-HSMM Auxiliary Variable Sampler

## B.1 Overview

This appendix considers the problem of drawing samples from posterior distributions formed under a Dirichlet prior and a truncated multinomial likelihood, by which we mean a Multinomial likelihood function where we condition on one or more counts being zero a priori. An example is the distribution with density

$$p(\pi|m,\alpha) \propto \underbrace{\prod_i \pi_i^{\alpha_i - 1}}_{\text{prior}} \cdot \underbrace{\left(\prod_{i \neq 1} \left(\frac{\pi_i}{1-\pi_1}\right)^{m_{1i}}\right)\left(\prod_{i \neq 2} \left(\frac{\pi_i}{1-\pi_2}\right)^{m_{2i}}\right)}_{\text{likelihood}} \tag{B.1}$$

where $\pi \in \Delta := \left\{x \in \mathbb{R}_+^n : \sum_i x_i = 1\right\}$, $\alpha \in \mathbb{R}_+^n$, and $\mathbb{R}_+ = \{x \in \mathbb{R} : x \geq 0\}$. We say the likelihood function has two *truncated* terms because each term corresponds to a multinomial likelihood defined on the full parameter $\pi$ but conditioned on the event that observations with a certain label are removed from the data.

Sampling this posterior distribution is of interest in inference algorithms for hierarchical Bayesian models based on the Dirichlet distribution or the Dirichlet Process, particularly the sampling algorithm for the Hierarchical Dirichlet Process Hidden Semi-Markov Model (HDP-HSMM) of Chapter 3, which must draw samples from such a distribution.

We provide an auxiliary variable (or data augmentation) [109] sampling algorithm that is easy to implement, fast both to mix and to execute, and easily scalable to high dimensions. This appendix will explicitly work with the finite Dirichlet distribution, but the sampler immediately generalizes to the Dirichlet Process case based on the Dirichlet

Process's definition in terms of the finite Dirichlet distribution and the Komolgorov extension theorem [87].

Section B.2 explains the problem in greater detail. Section B.3 provides a derivation of our sampling algorithm. Finally, Section B.4 provides numerical experiments in which we demonstrate the algorithm's significant advantages over a generic Metropolis-Hastings sampling algorithm.

Sampler code and functions to generate each plot in this appendix are available at `https://github.com/mattjj/dirichlet-truncated-multinomial`.

## ■ B.2 Problem Description

We say a vector $\pi \in \Delta$ is Dirichlet-distributed with parameter vector $\alpha \in \mathbb{R}^n_+$ if it has a density

$$p(\pi|\alpha) = \frac{\Gamma\left(\sum_{i=1}^n \alpha_i\right)}{\prod_{i=1}^n \Gamma(\alpha_i)} \prod_{i=1}^n \pi_i^{\alpha_i - 1} \tag{B.1}$$

$$=: \mathrm{Dir}(\pi|\alpha) \tag{B.2}$$

with respect to Lebesgue measure. The Dirichlet distribution and its generalization to arbitrary probability spaces, the Dirichlet Process, are common in Bayesian statistics and machine learning models. It is most often used as a prior over finite probability mass functions, such as the faces of a die, and paired with the multinomial likelihood, to which it is conjugate, viz.

$$\mathrm{Dir}(\pi|\alpha) \cdot \mathrm{Mult}(m|\pi) \propto \prod_i \pi_i^{\alpha - 1} \cdot \prod_i \pi_i^{m_i} \tag{B.3}$$

$$\propto \prod_i \pi_i^{\alpha_i + m_i - 1} \tag{B.4}$$

$$\propto \mathrm{Dir}(\pi|\alpha + m). \tag{B.5}$$

That is, given a count vector $m \in \mathbb{N}^n_+$, the posterior distribution is also Dirichlet with an updated parameter vector and, therefore, it is easy to draw samples from the posterior.

However, we consider a modified likelihood function which does not maintain the convenient conjugacy property: the *truncated* multinomial likelihood, which corresponds to deleting a particular set of counts from the count vector $m$ or, equivalently, conditioning on the event that they are not observed. The truncated multinomial likelihood where the first component is truncated can be written

$$\text{TruncMult}_{\{1\}}(m|\pi) := \prod_{i \neq 1} \left( \frac{\pi_i}{1 - \pi_1} \right)^{m_i} \tag{B.6}$$

$$= \text{Mult}(m|\pi, \{m_1 = 0\}). \tag{B.7}$$

In general, any subset of indices may be truncated; if a set $\mathcal{I} \subseteq \{1, 2, \ldots, n\}$ is truncated, then we write

$$\text{TruncMult}_{\mathcal{I}}(m|\pi) := \left( \frac{1}{1 - \sum_{i \in \mathcal{I}} \pi_i} \right)^{m_\cdot} \prod_{i \notin \mathcal{I}} \pi_i^{m_i} \tag{B.8}$$

where $m_\cdot = \sum_i m_i$.

In the case where the posterior is proportional to a Dirichlet prior and a single truncated multinomial likelihood term, the posterior is still simple to write down and sample. In this case, we may split the Dirichlet prior over $\mathcal{I}$ and its complement $\overline{\mathcal{I}} := \{1, 2, \ldots, n\} \setminus \mathcal{I}$; the factor over $\overline{\mathcal{I}}$ is conjugate to the likelihood, and so the posterior can be written

$$\text{Dir}(\pi|\alpha)\text{TruncMult}_{\mathcal{I}}(m|\pi) \propto \text{Dir}\left( \frac{\pi_{\mathcal{I}}}{1 - \sum_{i \in \mathcal{I}} \pi_i} \middle| \alpha_{\mathcal{I}} \right) \text{Dir}\left( \frac{\pi_{\overline{\mathcal{I}}}}{1 - \sum_{i \in \mathcal{I}} \pi_i} \middle| \alpha_{\overline{\mathcal{I}}} + m_{\overline{\mathcal{I}}} \right) \tag{B.9}$$

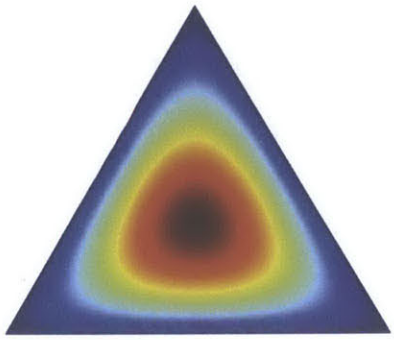
from which we can easily sample. However, given two or more truncated likelihood terms with different truncation patterns, no simple conjugacy property holds, and so it is no longer straightforward to construct samples from the posterior. For a visual comparison in the $n = 3$ case, see Figure B.1.

For the remainder of this appendix, we deal with the case where there are two likelihood terms, each with one component truncated. The generalization of the equations and algorithms to the case where any set of components is truncated is immediate.

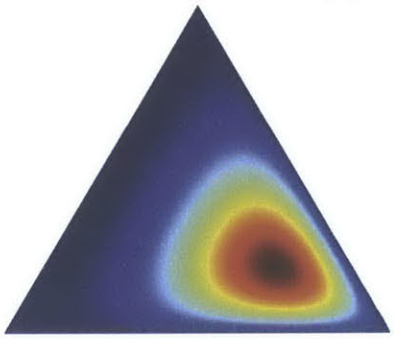
## ■ B.3 An Auxiliary Variable Sampler

Data augmentation methods are auxiliary variable methods that often provide excellent sampling algorithms because they are easy to implement and the component steps are simply conjugate Gibbs sampling steps, resulting in fast mixing. For an overview, see the survey [109].

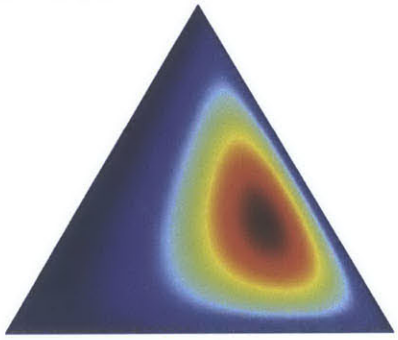
We can derive an auxiliary variable sampler for our problem by augmenting the distribution with geometric random variables $k = (k_1, k_2) = (\{k_{1j}\}, \{k_{2j}\})$. That is, we



**(a)** The prior, $\mathrm{Dir}(\pi|\alpha)$.



**(b)** The posterior proportional to $\mathrm{Dir}(\pi|\alpha) \cdot \mathrm{Mult}(m|\pi)$.



**(c)** The posterior proportional to $\mathrm{Dir}(\pi|\alpha) \cdot \mathrm{TruncMult}_{\{1\}}(m|\pi)$.

**Figure B.1:** Projected visualizations of the prior distribution $\mathrm{Dir}(\pi|\alpha)$ for $n = 3$ and $\alpha = (2, 2, 2)$ and the associated posterior distributions when paired with $\mathrm{Mult}(m|\pi)$ and $\mathrm{TruncMult}_{\{1\}}(m|\pi)$ where $m = (0, 2, 0)$. In low dimensions, the posteriors can be computed via direct numerical integration over a discretized mesh.

define for $k_{ij} = \{0, 1, 2, \ldots\}$ a new distribution $q$ such that

$$q(\pi, m, k|\alpha) \propto \left(\prod_i \pi_i^{\alpha-1}\right)\left(\prod_{i\neq 1} \pi_i^{m_{1i}}\right)\left(\prod_{i\neq 2} \pi_i^{m_{2i}}\right)\left(\prod_{j=1}^{m_{1\cdot}} \pi_1^{k_{1j}}\right)\left(\prod_{j=1}^{m_{2\cdot}} \pi_2^{k_{2j}}\right) \tag{B.1}$$

where $\{m_{1i}\}$ and $\{m_{2i}\}$ are sample counts corresponding to each likelihood, respectively, and $m_{i\cdot} := \sum_j m_{ij}$. Note that if we sum over all the auxiliary variables $k$, we have

$$\sum_k q(\pi, m, k|\alpha) \propto \left(\prod_i \pi_i^{\alpha-1}\right)\left(\prod_{i\neq 1} \pi_i^{m_{1i}}\right)\left(\prod_{i\neq 2} \pi_i^{m_{2i}}\right)\left(\prod_j \sum_{k_{1j}} \pi_1^{k_{1j}}\right)\left(\prod_j \sum_{k_{2j}} \pi_2^{k_{2j}}\right) \tag{B.2}$$

$$= \prod_i \pi_i^{\alpha-1}\left(\prod_{i\neq 1}\left(\frac{\pi_i}{1-\pi_1}\right)^{m_{1i}}\right)\left(\prod_{i\neq 2}\left(\frac{\pi_i}{1-\pi_2}\right)^{m_{2i}}\right) \tag{B.3}$$

$$\propto p(\pi, m|\alpha) \tag{B.4}$$

and so if we can construct samples of $\pi, k|m, \alpha$ from the distribution $q$ then we can form samples of $\pi|m, \alpha$ according to $p$ by simply ignoring the values sampled for $k$.

We construct samples of $\pi, k|m, \alpha$ by iterating Gibbs steps between $k|\pi, m, \alpha$ and $\pi|k, m, \alpha$. We see from (B.1) that each $k_{ij}$ in $k|\pi, m, \alpha = k|\pi, m$ is independent and distributed according to

$$q(k_{ij}|\pi, m) = (1-\pi_i)\pi_i^{k_{ij}}. \tag{B.5}$$

Therefore, each $k_{ij}$ follows a geometric distribution with success parameter $(1-\pi_i)$.

The distribution of $\pi|k, m, \alpha$ in $q$ is also simple:

$$q(\pi|m, k, \alpha) \propto \left(\prod_i \pi_i^{\alpha_i-1}\right)\left(\prod_{i\neq 1}\left(\frac{\pi_i}{1-\pi_1}\right)^{m_{1i}}\right)\left(\prod_{i\neq 2}\left(\frac{\pi_i}{1-\pi_2}\right)^{m_{2i}}\right) \tag{B.6}$$

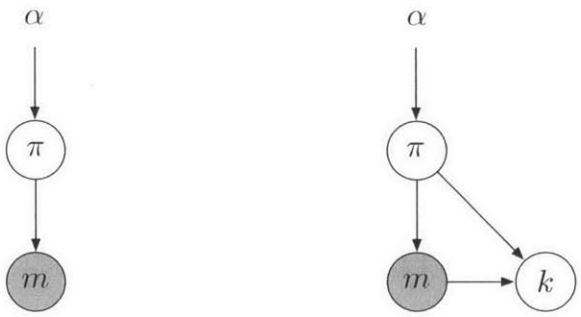
$$\cdot \left(\prod_{j=1}^{m_{1\cdot}}(1-\pi_1)\pi_1^{k_{1j}}\right)\left(\prod_{j=1}^{m_{2\cdot}}(1-\pi_2)\pi_2^{k_{2j}}\right) \tag{B.7}$$

$$\propto \mathrm{Dir}(\pi|\alpha + \tilde{m}) \tag{B.8}$$

where $\tilde{m}$ is a set of *augmented* counts including the values of $k$. In other words, the Dirichlet prior is conjugate to the augmented model. Therefore we can cycle through Gibbs steps in the augmented distribution and hence easily produce samples from the desired posterior. For a graphical model of the augmentation, see Figure B.2.

## ■ B.4 Numerical Experiments

In this section we perform several numerical experiments to demonstrate the advantages provided by the auxiliary variable sampler. We compare to a generic Metropolis-Hastings sampling algorithm. For all experiments, when a statistic is computed in terms




**(a)** Un-augmented distribution.

**(b)** Augmented distribution.

**Figure B.2:** Graphical models for the un-augmented and augmented probability models.

of a sampler chain's samples up to sample index $t$, we discard the first $\lfloor \frac{t}{2} \rfloor$ samples and use the remaining samples to compute the statistic.

**Metropolis-Hastings Sampler** We construct an MH sampling algorithm by using the proposal distribution which proposes a new position $\pi'$ given the current position $\pi$ via the proposal distribution

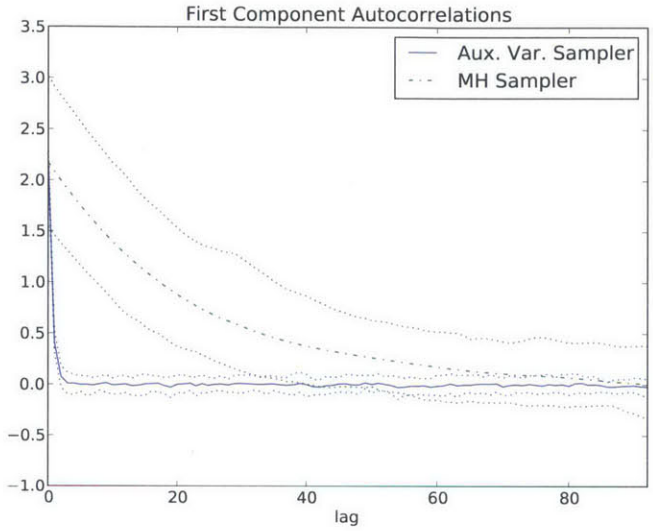
$$p(\pi'|\pi;\beta) = \mathrm{Dir}(\pi'|\beta \cdot \pi) \tag{B.1}$$

where $\beta > 0$ is a tuning parameter. This proposal distribution has several valuable properties:

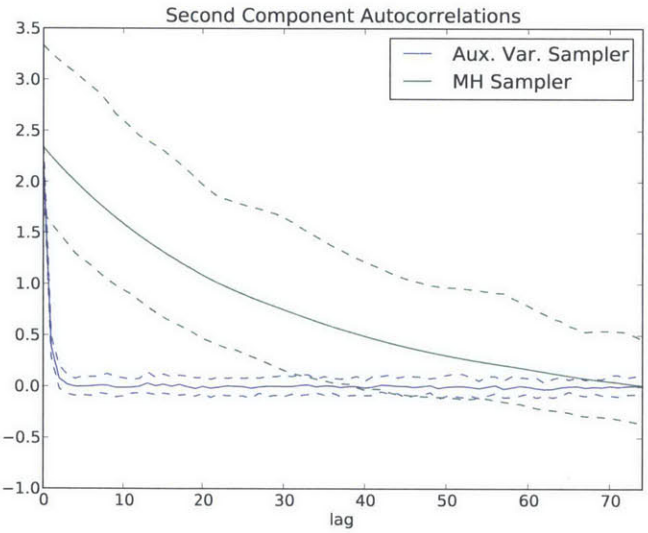
1. the mean and mode of the proposals are both $\pi$;
2. the parameter $\beta$ directly controls the concentration of the proposals, so that larger $\beta$ correspond to more local proposal samples;
3. the proposals are naturally confined to the support of the target distribution, while alternatives such as local Gaussian proposals would not satisfy the MH requirement that the normalization constant of the proposal kernel be constant for all starting points.

In our comparison experiments, we tune $\beta$ so that the acceptance probability is approximately 0.24.

**Sample Chain Autocorrelation** In Figure B.3 we compare the sample autocorrelation of the auxiliary variable sampler and the alternative MH sampler for several lags with $n = 10$. The reduced autocorrelation that is typical in the auxiliary variable sampler chain is indicative of faster mixing.




**(a)** Autocorrelations in the first (truncated) component.




**(b)** Autocorrelations in the second component.

**Figure B.3:** Autocorrelations for the auxiliary variable sampler and MH sampler chains with $\alpha_i = 2$, $n = 10$, $\beta = 160$. The solid lines show the mean autocorrelation over 50 randomly-initialized runs for each sampler, and the dashed lines show the 10th and 90th percentile autocorrelation chains over those runs. These plots can be reproduced with the function `autocorrelation` in `figures.py`.

**The $\widehat{R}$ Multivariate Potential Scale Reduction Factor** The $\widehat{R}$ statistic, also called the Multivariate Potential Scale Reduction Factor (MPSRF), was introduced in [16] and is a natural generalization of the scalar Scale Reduction Factor, introduced in [37] and discussed in [36, p. 296]. As a function of multiple independent sampler chains, the statistic compares the between-chain sample covariance matrix to the within-chain sample covariance matrix to measure mixing; good mixing is indicated by empirical convergence to the statistic's asymptotic value of unity.

Specifically, loosely following the notation of [16], with $\psi_{jt}^{(i)}$ for denoting the $i$th element of the parameter vector in chain $j$ at time $t$ (with $i = 1, \ldots, n$, $j = 1, \ldots, M$, and $t = 1, \ldots, T$), to compute the $n$-dimensional MPSRF we form

$$\widehat{V} = \frac{T-1}{T} W + \left(1 + \frac{1}{M}\right) B/T \tag{B.2}$$

where

$$W = \frac{1}{M(T-1)} \sum_{j=1}^{M} \sum_{t=1}^{T} (\psi_{jt} - \bar{\psi}_{j\cdot})(\psi_{jt} - \bar{\psi}_{j\cdot})^{\mathsf{T}} \tag{B.3}$$

$$B/T = \frac{1}{M-1} \sum_{j=1}^{M} (\bar{\psi}_{j\cdot} - \bar{\psi}_{\cdot\cdot})(\bar{\psi}_{j\cdot} - \bar{\psi}_{\cdot c\cdot})^{\mathsf{T}}. \tag{B.4}$$

The MPSRF itself is then defined when $W$ is full-rank as [16, Eq. 4.1 and Lemma 1]

$$\widehat{R} := \sup_{v \in \mathbb{R}^n} \frac{v^{\mathsf{T}} \widehat{V} v}{v^{\mathsf{T}} W v} \tag{B.5}$$

$$= \lambda_{\max}\left(W^{-1} \widehat{V}\right) \tag{B.6}$$

$$= \lambda_{\max}\left(W^{-\frac{1}{2}} \widehat{V} W^{\frac{1}{2}}\right) \tag{B.7}$$

where $\lambda_{\max}(A)$ denotes the eigenvalue of largest modulus of the matrix $A$ and the last line follows because conjugating by $W^{\frac{1}{2}}$ is a similarity transformation. Equivalently (and usefully for computation), we must find the largest solution $\lambda$ to $\det(\lambda W - \widehat{V}) = 0$.

However, as noted in [16, p. 446], the measure is incalculable when $W$ is singular, and because our samples are constrained to lie in the simplex in $n$ dimensions, the matrices involved will have rank $n-1$. Therefore when computing the $\widehat{R}$ statistic, we simply perform the natural Euclidean orthogonal projection to the $(n-1)$-dimensional affine subspace on which our samples lie; specifically, we define the statistic in terms of

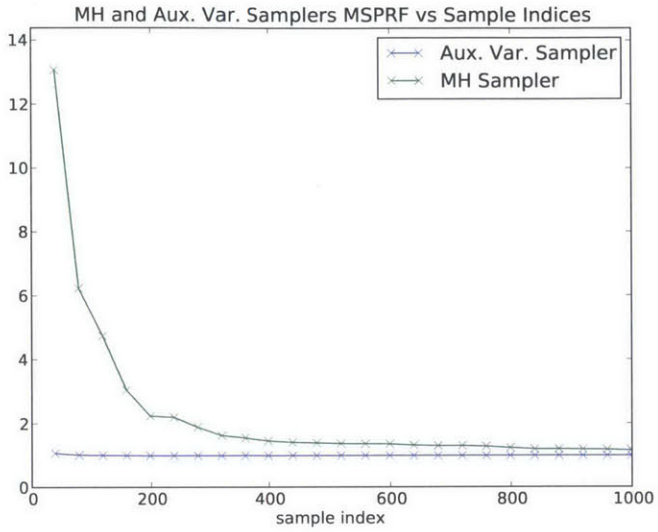
$Q^{\mathsf{T}}\widehat{V}Q$ and $Q^{\mathsf{T}}WQ$, where $Q$ is an $n \times (n-1)$ matrix such that $Q^{\mathsf{T}}Q = I_{(n-1)}$ and

$$QR = \begin{pmatrix} -(n-1) & 1 & \cdots & 1 & 1 \\ 1 & -(n-1) & \cdots & 1 & 1 \\ & & \vdots & & \\ 1 & 1 & \cdots & 1 & -(n-1) \\ 1 & 1 & \cdots & 1 & 1 \end{pmatrix} \tag{B.8}$$

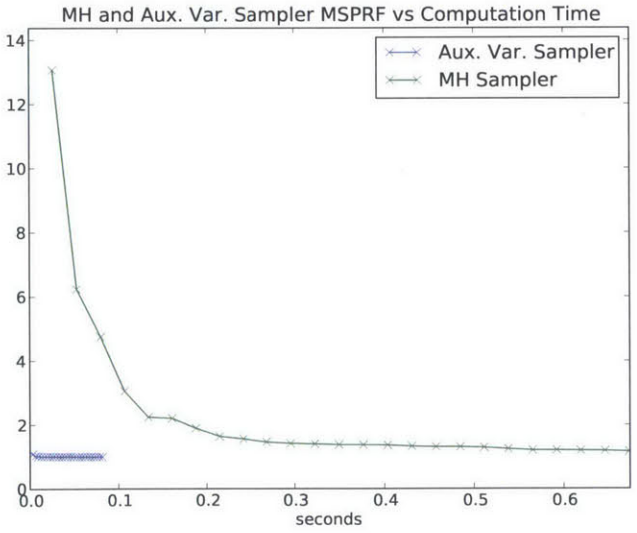
for upper-triangular $R$ of size $(n-1) \times (n-1)$.

Figure B.4 shows the MPSRF of both samplers computed over 50 sample chains for $n = 10$ dimensions, and Figure B.5 shows the even greater performance advantage of the auxiliary variable sampler in higher dimensions.

**Statistic Convergence** Finally, we show the convergence of the component-wise mean and variance statistics for the two samplers. We estimated the true statistics by forming estimates using samples from 50 independent chains each with 5000 samples, effectively using 250000 samples to form the estimates. Next, we plotted the $\ell_2$ distance between these "true" statistic vectors and those estimated at several sample indices along the 50 runs for each of the sampling algorithms. See the plots in Figure B.6.

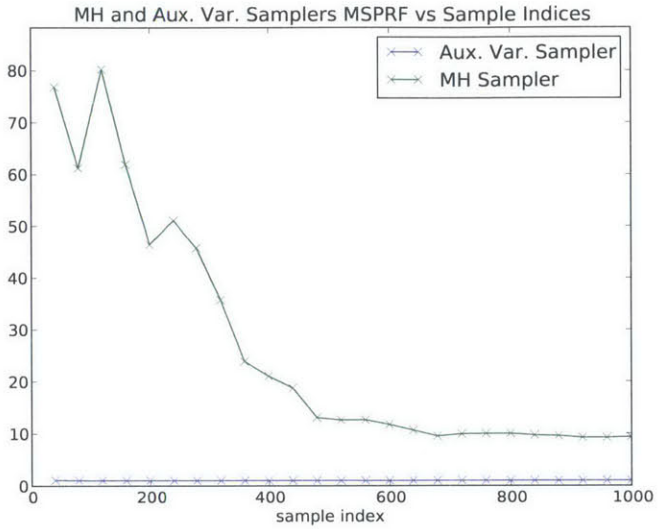



**(a)** The horizontal axis is the sample index.

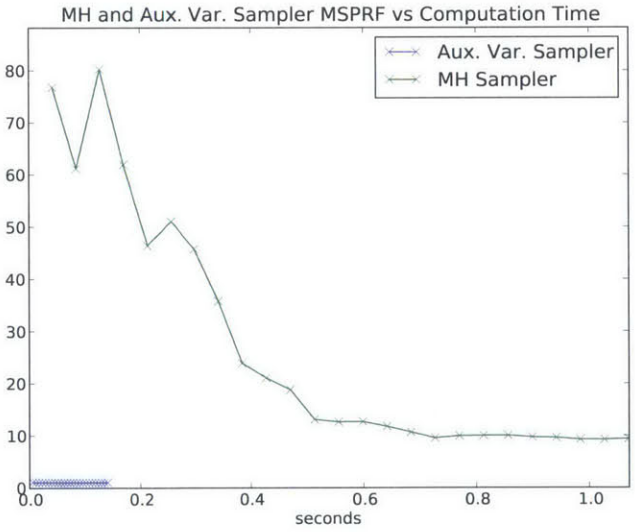



**(b)** The horizontal axis is elapsed time.

**Figure B.4:** The $\widehat{R}$ Multivariate Potential Scale Reduction Factor [16] for the auxiliary variable sampler and MH sampler with $\alpha_i = 2$, $n = 10$, and $\beta = 160$, with horizontal axes scaled by sample index and elapsed time. For each sampler, 5000 samples were drawn for each of 50 randomly-initialized runs, and the MPSRF was computed at 25 equally-spaced intervals. These plots can be reproduced with the function `Rhatp` in `figures.py`.

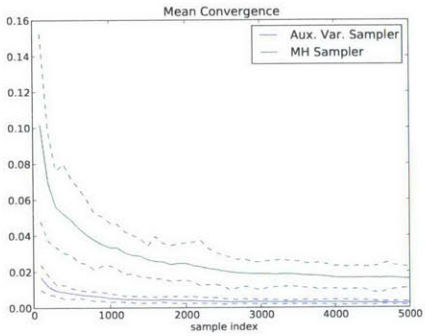



**(a)** The horizontal axis is the sample index.

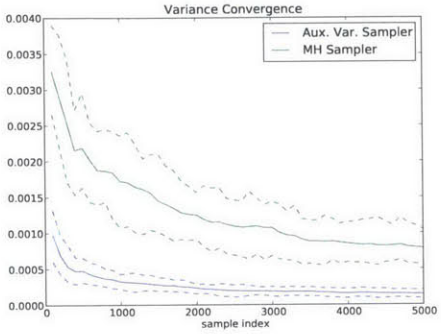



**(b)** The horizontal axis is elapsed time.

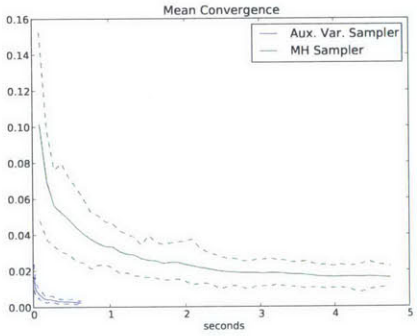
**Figure B.5:** The $\widehat{R}$ Multivariate Potential Scale Reduction Factor [16] for the auxiliary variable sampler and MH sampler with $\alpha_i = 2$, $n = 20$, and $\beta = 160$, with horizontal axes scaled by sample index and elapsed time. For each sampler, 5000 samples were drawn for each of 50 randomly-initialized runs, and the MPSRF was computed at 25 equally-spaced intervals. These plots can be reproduced with the function `Rhatp` in `figures.py`.



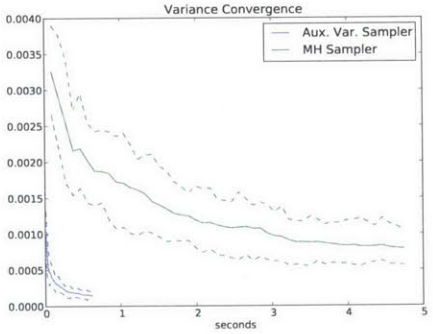

(a) $\ell_2$ error of component-wise mean estimate vs sample index.




(b) $\ell_2$ error of component-wise variance estimate vs sample index.




(c) $\ell_2$ error of component-wise mean estimate vs elapsed time.




(d) $\ell_2$ error of component-wise variance estimate vs elapsed time.

**Figure B.6:** Component-wise statistic convergence for the auxiliary variable sampler and MH sampler with $\alpha_i = 2$, $n = 10$, and $\beta = 160$, with horizontal axes scaled by sample index and elapsed time. For each sampler, 5000 samples were drawn for each of 50 randomly-initialized runs. The $\ell_2$ distances from estimated "true" parameters are plotted, with the solid lines corresponding to the mean error and the dashed lines corresponding to 10th and 90th percentile errors. These plots can be reproduced with the function `statistic_convergence` in `figures.py`.

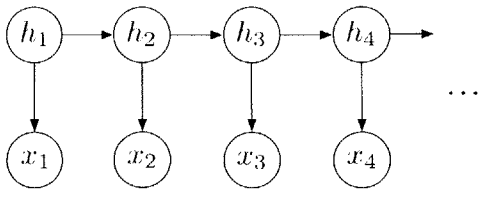
Appendix C

# Spectral Learning of HMM Predictive State Representations

This appendix is a summary explanation of the algorithm in "A Spectral Algorithm for Learning Hidden Markov Models" (COLT 2009), though there may be some slight notational inconsistencies with the original paper.

The idea is to maintain output predictions in a recursive inference algorithm, instead of the usual method of maintaining hidden state predictions, and to represent the HMM only in terms of the maps necessary to update output predictions given new data. This approach limits the inference computations the algorithm can perform (it can't answer any queries about the hidden states since it doesn't explicitly deal with them at all), but it also reduces the complexity of the model parameters that are learned and thus makes learning easier. The learning algorithm uses an SVD and matrix operations, so it avoids the local-optima problems of EM or any other algorithms based on maximizing data likelihood over the usual HMM parameterization. The COLT paper includes error bounds and analysis.

**Notation.** *For a vector $v$ in a subspace $\mathcal{V} \subseteq \mathbb{R}^k$ and a matrix $C$ with linearly independent columns and* $\text{range}(C) \supseteq \mathcal{V}$ *we use $[v]^C$ to denote the coordinate vector of $v$ relative to the ordered basis given by the columns of $C$, and $[v]$ or simply $v$ to denote the coordinate vector of $v$ relative to the standard basis of $\mathbb{R}^k$. Similarly, for a linear map $\mathcal{A} : \mathcal{V} \to \mathcal{V}$ we use $[\mathcal{A}]_C^C$ to denote the matrix of $\mathcal{A}$ relative to domain and codomain bases given by the columns of $C$, and $[\mathcal{A}]$ to indicate its matrix relative to standard bases. For a matrix $A$ we also use $[A]_{ij}$ to denote the $(i,j)$th entry.*



**Definition C.1** (Hidden Markov Model). *A time-homogeneous, discrete* Hidden Markov Model (HMM) *is a probability distribution on random variables* $\{(x_t, h_t)\}_{t\in\mathbb{N}}$ *satisfying the conditional independences implied by the graphical model, where* $\text{range}(h_t) = [m] := \{1, 2, \ldots, m\}$ *and* $\text{range}(x_t) = [n]$ *where* $n \geq m$. *The* standard parameterization *is the triple* $(T, O, \pi)$, *where*

$$\begin{aligned} T \in \mathbb{R}^{m\times m}, &\quad [T]_{ij} = \Pr[h_t = i | h_{t-1} = j] \\ O \in \mathbb{R}^{n\times m}, &\quad [O]_{ij} = \Pr[x_t = i | h_t = j] \\ \pi \in \mathbb{R}^{m}, &\quad [\pi]_j = \Pr[h_1 = j]. \end{aligned}$$

*We assume* $T$ *and* $O$ *to have full column rank and* $[\pi]_j > 0\ \forall j \in [m]$.

**Definition C.2** (Observation Prediction). *An* observation prediction *for any time* $t$ *is a vector* $\vec{x}_t \in \mathbb{R}^n$ *defined in the standard basis by*

$$[\vec{x}_t]_i := \Pr[x_t = i | x_{1:t-1} = \bar{x}_{1:t-1}] \tag{C.1}$$

*for some fixed (implicit) sequence* $\bar{x}_{1:t-1}$.

**Claim C.1.** *Every observation prediction* $\vec{x}_t$ *lies in a subspace* $\mathcal{U} := \text{range}(O) \subseteq \mathbb{R}^n$ *with* $\dim(\mathcal{U}) = m$.

*Proof.* By the conditional independences of the HMM, for any $t$ we have

$$\begin{aligned} \Pr[x_t = i | x_{1:t-1} = \bar{x}_{1:t-1}] = &\sum_{j\in[m]} \Pr[x_t = i | h_t = j] \\ &\cdot \Pr[h_t = j | x_{1:t-1} = \bar{x}_{1:t-1}] \end{aligned} \tag{C.2}$$

so therefore we can write

$$\vec{x}_t = O\vec{h}_t, \quad [\vec{h}_t]_j := \Pr[h_t = j | x_{1:t-1} = \bar{x}_{1:t-1}]. \tag{C.3}$$

Equivalently, we can say $[\vec{x}_t]^O = \vec{h}_t$. See Figure C.1. □

**Claim C.2.** *The observation prediction subspace* $\mathcal{U}$ *satisfies* $\mathcal{U} = \text{range}(P_{2,1})$, *where* $[P_{2,1}]_{ij} := \Pr[x_2 = i, x_1 = j]$

*Proof.* We can write the joint distribution over $(x_1, x_2)$ as

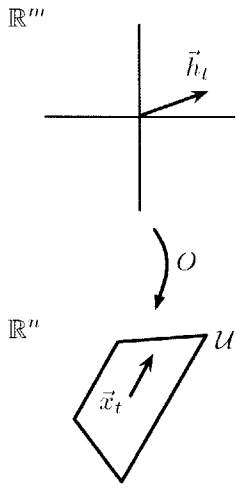



**Figure C.1:** We can view $O$ as (the matrix of) a map from hidden state beliefs to output predictions (with respect to the standard bases). Not shown is the fact that both $\vec{h}_t$ and $\vec{x}_t$ lie in the simplices of $\mathbb{R}^m$ and $\mathbb{R}^n$, respectively, and that $O$ maps the simplex in $\mathbb{R}^m$ to (a subset of) the simplex in $\mathbb{R}^n$.

$$\Pr[x_2 = i, x_1 = j] = \sum_{\bar{h}_1} \sum_{\bar{h}_2} \Pr[x_2 = i, x_1 = j, h_1 = \bar{h}_1, h_2 = \bar{h}_2] \tag{C.4}$$

$$= \sum_{\bar{h}_1} \Pr[h_1 = \bar{h}_1] \Pr[x_1 = j | h_1 = \bar{h}_1]$$

$$\cdot \sum_{\bar{h}_2} \Pr[h_2 = \bar{h}_2 | h_1 = \bar{h}_1] Pr[x_2 = i | h_2 = \bar{h}_2] \tag{C.5}$$

and we can write that sum as $P_{2,1} = OT \operatorname{diag}^*(\pi) O^\mathsf{T}$, where $\operatorname{diag}^*(\cdot)$ maps a vector to a diagonal matrix in the usual way. By our rank and positivity assumptions, we see that $P_{2,1}$ satisfies $\operatorname{range}(P_{2,1}) = \operatorname{range}(O) = \mathcal{U}$. □

We can directly estimate $P_{2,1}$ with empirical statistics, and as a consequence of Claim C.2 we can then get a basis for $\mathcal{U}$ by using an SVD of $P_{2,1}$. Note that we aren't getting an estimate of $O$ this way, but just its column space.

**Claim C.3.** *Given an observation $x_t = \bar{x}_t$, there is a linear map $\mathcal{B}_{\bar{x}_t} : \mathcal{U} \to \mathcal{U}$ such that*

$$\mathcal{B}_{\bar{x}_t}(\vec{x}_t) = \alpha \vec{x}_{t+1} \tag{C.6}$$

*for some $\alpha = \alpha(\bar{x}_t, \vec{x}_t)$, a scalar normalization factor chosen to ensure $1^\mathsf{T} \vec{x}_{t+1} = 1$.*

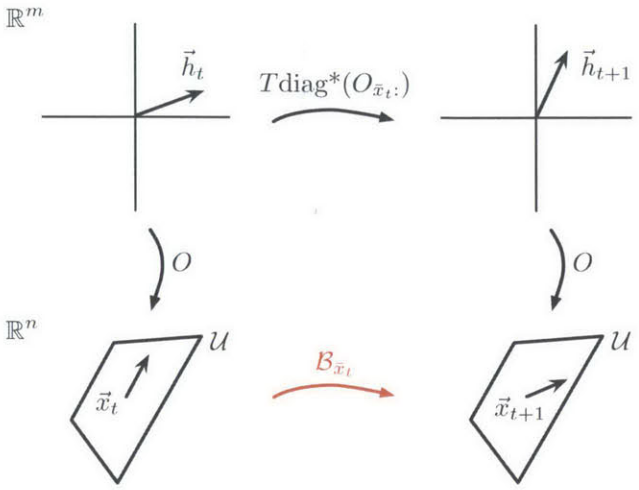



**Figure C.2:** The matrix $T\,\text{diag}^*(O_{\bar{x}_t:})$ is the matrix (relative to standard bases) of a linear map that updates hidden state beliefs given an observation $\bar{x}_t$, up to renormalization which the figure does not show. The linear map $\mathcal{B}_{\bar{x}_t}$ is the update for output predictions.

*Proof.* Following the usual recursive update for HMM forward messages, we have

$$\Pr[h_{t+1} = i, x_{1:t} = \bar{x}_{1:t}] = \sum_{j \in [m]} \underbrace{\Pr[h_{t+1} = i | h_t = j]}_{[T]_{ij}} \cdot \underbrace{\Pr[x_t = \bar{x}_t | h_t = j]}_{[O]_{\bar{x}_t j}} \cdot \underbrace{\Pr[h_t = j, x_{1:t-1} = \bar{x}_{1:t-1}]}_{\propto [\vec{x}_t]_j^O} . \tag{C.7}$$

Therefore we can write the map $\mathcal{B}_{\bar{x}_t}$ as an $m \times m$ matrix relative to the basis for $\mathcal{U}$ given by the columns of $O$:

$$[\mathcal{B}_{\bar{x}_t}]_O^O = T\,\text{diag}^*(O_{\bar{x}_t:}) \tag{C.8}$$

where $O_{k:}$ denotes the vector formed by the $k$th row of $O$. We require $1^\mathsf{T} \vec{x}_{t+1} = 1$, so we have $\alpha = O_{\bar{x}_t:}^\mathsf{T} \vec{x}_t$.

See Figure C.2. □

**Note.** *Renormalization always works because $T\,\text{diag}^*(O_{\bar{x}_t:})$ preserves the non-negative orthant of $\mathbb{R}^m$, and $O$ maps the simplex in $\mathbb{R}^m$ to (a subset of) the simplex in $\mathbb{R}^n$. The orthant preservation properties of these maps are an immediate consequence of the fact that the matrices (with respect to standard bases) are entry-wise non-negative. In fact,*

*instead of tracking vectors, we should be tracking rays in the non-negative orthant.*

For any new observation $x$, the map $\mathcal{B}_x$ implements the "belief update" on output predictions, up to a normalization factor which we can compute on the fly via $1^\top \vec{x}_t = 1$. Note that we can also write $\mathcal{B}_x$ as an $n \times n$ matrix relative to the standard basis of $\mathbb{R}^n$:

$$[\mathcal{B}_x] = OT \operatorname{diag}^*(O_{x:})O^\dagger \tag{C.9}$$

where $O^\dagger := (O^\top O)^{-1}O^\top$ is the pseudoinverse of $O$. Recall $\vec{x}_t = O\vec{h}_t$ and hence $\vec{h}_t = O^\dagger \vec{x}_t$.

We would like to write $\mathcal{B}_x$ as a matrix without reference to the standard HMM parameters $(T, O, \pi)$, since we want to avoid learning them at all.

**Claim C.4.** *Let $U \in \mathbb{R}^{n \times m}$ be a matrix whose columns form an orthonormal basis for $\mathcal{U}$. We can write $\mathcal{B}_x$ as a matrix relative to the standard basis of $\mathbb{R}^n$ as*

$$[\mathcal{B}_x] = P_{3,x,1}U(U^\top P_{2,1}U)^{-1}U^\top \tag{C.10}$$

*where*

$$[P_{3,x,1}]_{ij} := \Pr[x_3 = i, x_2 = x, x_1 = j] \; \forall x \in [n]. \tag{C.11}$$

*Proof.* We can express the matrix $P_{3,x,1}$ in a form similar to that for the matrix $[\mathcal{B}_x]$ in Equation (C.9):

$$P_{3,x,1} = \underbrace{OT \operatorname{diag}^*(O_{x:})O^\dagger}_{[\mathcal{B}_x]} P_{2,1}. \tag{C.12}$$

Intuitively, we want to remove the $P_{2,1}$ on the right, since that would give us $[\mathcal{B}_x]$ in terms of quantities we can readily estimate, but we cannot form the inverse of $P_{2,1}$ because it is $n \times n$ and has rank $m \leq n$. However, $P_{2,1}$ has row and column space $\mathcal{U}$ (intuitively, its restriction to $\mathcal{U}$ is invertible), thus we can substitute

$$P_{2,1} = U(U^\top P_{2,1}U)U^\top \tag{C.13}$$

to get

$$P_{3,x,1} = OT \operatorname{diag}^*(O_{x:}) O^\dagger U (U^\mathsf{T} P_{2,1} U) U^\mathsf{T} \tag{C.14}$$

and hence

$$P_{3,x,1} U (U^\mathsf{T} P_{2,1} U)^{-1} U^\mathsf{T} = OT \operatorname{diag}^*(O_{x:}) O^\dagger = [\mathcal{B}_x]. \tag{C.15}$$

□

Because we can estimate each $P_{3,x,1}$ as well as $P_{2,1}$ from data by empirical statistics, and we can obtain a $U$ using an SVD, we can now estimate a representation of $\mathcal{B}_x$ from data using the expression in Claim C.4. We can also directly estimate $P_1$ from empirical statistics, where $[P_1]_i := \Pr[x_1 = i]$, and hence we can use these estimated quantities to recursively compute $\Pr[x_t | x_{1:t-1}]$ and $\Pr[x_{1:t-1}]$ given observations up to and including time $t-1$.

Since $\dim(U) = m \leq n$, we can use the columns of $U$ as our basis for $\mathcal{U}$ to get a more economical coordinate representation of $\vec{x}_t$ and $\mathcal{B}_x$ than in the standard basis of $\mathbb{R}^n$:

**Definition C.3** (HMM PSR Representation). *For any fixed $U \in \mathbb{R}^{n \times m}$ with $\operatorname{range}(U) = \mathcal{U}$ and $U^\mathsf{T} U = I_{m \times m}$, we define the* belief vector *at time $t$ by*

$$\vec{b}_t := [\vec{x}_t]^U = U^\mathsf{T} \vec{x}_t \in \mathbb{R}^m \tag{C.16}$$

*and in particular for $t = 1$ we have*

$$\vec{b}_1 = U^\mathsf{T} O \pi. \tag{C.17}$$

*For each possible observation $x \in [n]$, we define the matrix $B_x \in \mathbb{R}^{m \times m}$ by*

$$B_x := [\mathcal{B}_x]^U_U = (U^\mathsf{T} O) T \operatorname{diag}^*(O_{x:}) (U^\mathsf{T} O)^{-1}. \tag{C.18}$$

*Finally, for normalization purposes, it is convenient to maintain the appropriate mapping of the ones (co-)vector, noting $1^\mathsf{T} O \vec{h} = 1$ if and only if $1^\mathsf{T} \vec{h} = 1$ because $1^\mathsf{T} O = 1^\mathsf{T}$:*

$$\vec{b}_\infty := [1]^U = U^\mathsf{T} 1 \in \mathbb{R}^m. \tag{C.19}$$

The box below summarizes the method for learning an HMM PSR representation from data and how to use an HMM PSR representation to perform some recursive inference computations.

**Learning**

$$\begin{aligned}
\widehat{U} &= \text{ThinSVD}(\widehat{P}_{2,1}) \\
\widehat{b}_1 &= \widehat{U}^\top \widehat{P}_1 \\
\widehat{B}_x &= \widehat{U}^\top \widehat{P}_{3,x,1} (\widehat{U}^\top \widehat{P}_{2,1})^\dagger \; \forall x \in [n] \\
\widehat{b}_\infty &= \widehat{U}^\top 1
\end{aligned}$$

**Inference**

$$\begin{aligned}
\Pr[x_{1:t}] &= \vec{b}_\infty^\top B_{x_{t:1}} \vec{b}_1 && \text{sequence probability} \\
\Pr[x_t | x_{1:t-1}] &= \vec{b}_\infty^\top B_{x_t} \vec{b}_t && \text{prediction} \\
\vec{b}_{t+1} &= \frac{B_{x_t} \vec{b}_t}{\vec{b}_\infty^\top B_{x_t} \vec{b}_t} && \text{recursive update}
\end{aligned}$$

# Bibliography


[1] S. Ahn, A. Korattikara, and M. Welling. "Bayesian Posterior Sampling via Stochastic Gradient Fisher Scoring". In: *International Conference on Machine Learning* (2012).

[2] S. Amari and H. Nagaoka. *Methods of Information Geometry.* Vol. 191. American Mathematical Society, 2007.

[3] Sanjeev Arora and Boaz Barak. *Computational Complexity: a Modern Approach.* Cambridge University Press, 2009.

[4] Richard Askey, Richard Askey, Richard Askey, and Richard Askey. *Orthogonal Polynomials and Special Functions.* SIAM, 1975.

[5] M. J. Beal, Z. Ghahramani, and C. E. Rasmussen. "The Infinite Hidden Markov Model". In: *Advances in Neural Information Processing Systems* 14 (2002), pp. 577–584.

[6] Matthew James Beal. "Variational Algorithms for Approximate Bayesian Inference". PhD thesis. University of London, 2003.

[7] R. Bekkerman, M. Bilenko, and J. Langford. *Scaling Up Machine Learning: Parallel and Distributed Approaches.* Cambridge University Press, 2012.

[8] Luca Benvenuti and Lorenzo Farina. "A Tutorial on the Positive Realization Problem". In: *Automatic Control, IEEE Transactions on* 49.5 (2004), pp. 651–664.

[9] A. Berman and R.J. Plemmons. "Nonnegative Matrices in the Mathematical Sciences". In: *Classics in Applied Mathematics,* 9 (1979).

[10] Dimitri P Bertsekas and John N Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods.* Prentice Hall, 1989.

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

[12] D.M. Blei, A.Y. Ng, and M.I. Jordan. "Latent Dirichlet Allocation". In: *the Journal of machine Learning research* 3 (2003), pp. 993–1022.

[13] Erik G Boman, Doron Chen, Ojas Parekh, and Sivan Toledo. "On Factor Width and Symmetric H-matrices". In: *Linear algebra and its applications* 405 (2005), pp. 239–248.

[14] Léon Bottou. "Online Learning and Stochastic Approximations". In: *On-line learning in neural networks* 17 (1998), p. 9.

[15] Tamara Broderick, Nicholas Boyd, Andre Wibisono, Ashia C Wilson, and Michael Jordan. "Streaming Variational Bayes". In: *Advances in Neural Information Processing Systems*. 2013, pp. 1727–1735.

[16] S. P. Brooks and A. Gelman. "General Methods for Monitoring Convergence of Iterative Simulations". In: *Journal of Computational and Graphical Statistics* (1998), pp. 434–455.

[17] Michael Bryant and Erik B Sudderth. "Truly Nonparametric Online Variational Inference for Hierarchical Dirichlet Processes". In: *Advances in Neural Information Processing Systems*. 2012, pp. 2708–2716.

[18] James Glass Chia-ying Lee Yu Zhang. "Joint Learning of Phonetic Units and Word Pronunciations for ASR". In: *Proceedings of the 2013 Conference on Empirical Methods on Natural Language Processing*. 2013.

[19] E. Çinlar. *Probability and Stochastics*. Springer, 2010.

[20] Mohammad Dahleh, Munther A. Dahleh, and George Verghese. "6.241 Dynamic Systems and Control Notes". Accessed 29 Apr, 2014. Fall 2008. URL: http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-241j-dynamic-systems-and-control-spring-2011/readings/.

[21] Steven Davis and Paul Mermelstein. "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences". In: *Acoustics, Speech and Signal Processing, IEEE Transactions on* 28.4 (1980), pp. 357–366.

[22] J.W. Demmel. *Applied Numerical Linear Algebra*. Society for Industrial Mathematics, 1997.

[23] M. Dewar, C. Wiggins, and F. Wood. "Inference in Hidden Markov Models with Explicit State Duration Distributions". In: *Signal Processing Letters, IEEE* 99 (2012), pp. 1–1.

[24] David Duvenaud, James Lloyd, Roger Grosse, Joshua Tenenbaum, and Ghahramani Zoubin. "Structure Discovery in Nonparametric Regression through Compositional Kernel Search". In: *Proceedings of The 30th International Conference on Machine Learning*. 2013, pp. 1166–1174.

[25] Lorenzo Farina and Sergio Rinaldi. *Positive Linear Systems: Theory and Applications*. Vol. 50. John Wiley & Sons, 2011.

[26] Paul Fearnhead. "Exact and Efficient Bayesian Inference for Multiple Changepoint Problems". In: *Statistics and computing* 16.2 (2006), pp. 203–213.

[27] Thomas S Ferguson. "A Bayesian Analysis of some Nonparametric Problems". In: *The Annals of Statistics* (1973), pp. 209–230.

[28] Shai Fine, Yoram Singer, and Naftali Tishby. "The Hierarchical Hidden Markov Model: Analysis and Applications". In: *Machine learning* 32.1 (1998), pp. 41–62.

[29] Colin Fox and Albert Parker. "Convergence in Variance of First-Order and Second-Order Chebyshev Accelerated Gibbs Samplers". 2013. URL: http://www.physics.otago.ac.nz/data/fox/publications/SIAM_CS_2012-11-30.pdf.

[30] E. B. Fox. "Bayesian Nonparametric Learning of Complex Dynamical Phenomena". Ph.D. Thesis. Cambridge, MA: MIT, 2009.

[31] 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.

[32] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky. "Sharing Features among Dynamical Systems with Beta Processes". In: *Neural Information Processing Systems 22*. MIT Press, 2010.

[33] Emily Fox, Erik B Sudderth, Michael I Jordan, and Alan S Willsky. "Bayesian Nonparametric Inference of Switching Dynamic Linear Models". In: *Signal Processing, IEEE Transactions on* 59.4 (2011), pp. 1569–1585.

[34] A. Frommer and D.B. Szyld. "Asynchronous Two-Stage Iterative Methods". In: *Numerische Mathematik* 69.2 (1994), pp. 141–153.

[35] A. Frommer and D.B. Szyld. "On Asynchronous Iterations". In: *Journal of computational and applied mathematics* 123.1 (2000), pp. 201–216.

[36] A. Gelman, J.B. Carlin, H.S. Stern, and D.B. Rubin. *Bayesian Data Analysis*. Second. CRC Press, 2004.

[37] A. Gelman and D.B. Rubin. "Inference from Iterative Simulation using Multiple Sequences". In: *Statistical science* 7.4 (1992), pp. 457–472.

[38] Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian Data Analysis*. CRC Press, 2013.

[39] Stuart Geman and Donald Geman. "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 6 (1984), pp. 721–741.

[40] Z. Ghahramani and M. I. Jordan. "Factorial Hidden Markov Models". In: *Machine Learning* 29.2 (1997), pp. 245–273.

[41] James R Glass. "A Probabilistic Framework for Segment-Based Speech Recognition". In: *Computer Speech & Language* 17.2 (2003), pp. 137–152.

[42] J. Gonzalez, Y. Low, A. Gretton, and C. Guestrin. "Parallel Gibbs Sampling: From Colored Fields to Thin Junction Trees". In: *In Artificial Intelligence and Statistics (AISTATS)*. Ft. Lauderdale, FL, May 2011.

[43] Peter J Green and Sylvia Richardson. "Modelling Heterogeneity With and Without the Dirichlet Process". In: *Scandinavian journal of statistics* 28.2 (2001), pp. 355–375.

[44] Thomas L Griffiths, Mark Steyvers, David M Blei, and Joshua B Tenenbaum. "Integrating Topics and Syntax". In: *Advances in neural information processing systems*. 2004, pp. 537–544.

[45] Roger Baker Grosse. "Model Selection in Compositional Spaces". PhD thesis. Massachusetts Institute of Technology, 2014.

[46] Roger Grosse, Ruslan R Salakhutdinov, William T Freeman, and Joshua B Tenenbaum. "Exploiting Compositionality to Explore a Large Space of Model Structures". In: *arXiv preprint arXiv:1210.4856* (2012).

[47] Y. Guédon. "Exploring the state sequence space for hidden Markov and semi-Markov chains". In: *Computational Statistics and Data Analysis* 51.5 (2007), pp. 2379–2409. ISSN: 0167-9473. DOI: http://dx.doi.org/10.1016/j.csda.2006.03.015.

[48] Yann Guédon. "Hidden Hybrid Markov/Semi-Markov Chains". In: *Computational statistics & Data analysis* 49.3 (2005), pp. 663–688.

[49] Trevor Hastie, Robert Tibshirani, Jerome Friedman, T Hastie, J Friedman, and R Tibshirani. *The Elements of Statistical Learning*. Springer, 2009.

[50] 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.

[51] Katherine A Heller, Yee W Teh, and Dilan Görür. "Infinite Hierarchical Hidden Markov Models". In: *International Conference on Artificial Intelligence and Statistics*. 2009, pp. 224–231.

[52] M Hoffman, D Blei, Chong Wang, and John Paisley. "Stochastic Variational Inference". In: *Journal of Machine Learning Research* 14 (2013), pp. 1303–1347.

[53] Matt D. Hoffman, David M. Blei, and F. Bach. "Online Learning for Latent Dirichlet Allocation". In: *Advances in Neural Information Processing Systems* 23 (2010), pp. 856–864.

[54] Nicolas H Hudson. "Inference in Hybrid Systems with Applications in Neural Prosthetics". PhD thesis. California Institute of Technology, 2008.

[55] A. Ihler and D. Newman. "Understanding Errors in Approximate Distributed Latent Dirichlet Allocation". In: *Knowledge and Data Engineering, IEEE Transactions on* 24.5 (2012), pp. 952–960.

[56] H. Ishwaran and M. Zarepour. "Markov Chain Monte Carlo in Approximate Dirichlet and Beta Two-Parameter Process Hierarchical Models". In: *Biometrika* 87.2 (2000), pp. 371–390.

[57] Hemant Ishwaran and Mahmoud Zarepour. "Exact and Approximate Sum Representations for the Dirichlet Process". In: *Canadian Journal of Statistics* 30.2 (2002), pp. 269–283.

[58] Fred Jelinek. "Speech Recognition by Statistical Methods". In: *Proceedings of the IEEE* 64 (1976), pp. 532–556.

[59] Matthew J Johnson. "Bayesian Nonparametric Learning with Semi-Markovian Dynamics". S.M. thesis. Massachusetts Institute of Technology, 2010.

[60] Matthew J. Johnson and Alan S. Willsky. "Bayesian Nonparametric Hidden Semi-Markov Models". In: *Journal of Machine Learning Research* 14 (Feb. 2013), pp. 673–701.

[61] Michael T Johnson. "Capacity and Complexity of HMM Duration Modeling Techniques". In: *Signal Processing Letters, IEEE* 12.5 (2005), pp. 407–410.

[62] Raphaël Jungers. "The Joint Spectral Radius". In: *Lecture Notes in Control and Information Sciences* 385 (2009).

[63] J. A. Kelner, L. Orecchia, A. Sidford, and Z. A. Zhu. *A Simple, Combinatorial Algorithm for Solving SDD Systems in Nearly-Linear Time*. 2013. arXiv: 1301.6628 [cs.DS].

[64] H. Kim, M. Marwah, M. Arlitt, G. Lyon, and J. Han. *Unsupervised Disaggregation of Low Frequency Power Measurements.* Tech. rep. HP Labs Tech. Report, 2010.

[65] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques.* MIT Press, 2009.

[66] J. Z. Kolter and M. J. Johnson. "REDD: A Public Data Set for Energy Disaggregation Research". In: *SustKDD Workshop on Data Mining Applications in Sustainability.* San Diego, California, USA, 2011.

[67] Brenden Lake, Chia-ying Lee, James Glass, and Joshua Tenenbaum. "One-shot Learning of Generative Speech Concepts". In: *Proceedings of the 36th Annual Meeting of the Cognitive Science Society.* 2014.

[68] Chia-ying Lee and James Glass. "A Nonparametric Bayesian Approach to Acoustic Model Discovery". In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1.* Association for Computational Linguistics. 2012, pp. 40–49.

[69] Li-wei H Lehman, Shamim Nemati, Ryan P Adams, and Roger G Mark. "Discovering Shared Dynamics in Physiological Signals: Application to Patient Monitoring in ICU". In: *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE.* IEEE. 2012, pp. 5939–5942.

[70] Percy Liang, Slav Petrov, Michael I Jordan, and Dan Klein. "The Infinite PCFG Using Hierarchical Dirichlet Processes". In: *EMNLP-CoNLL.* 2007, pp. 688–697.

[71] Martin Lindén, Stephanie Johnson, Jan-Willem van de Meent, Rob Phillips, and Chris H Wiggins. "Analysis of DNA Looping Kinetics in Tethered Particle Motion Experiments using Hidden Markov Models". In: *Biophysical Journal* 104.2 (2013), 418A–418A.

[72] Y. Liu, O. Kosut, and A. S. Willsky. "Sampling GMRFs by Subgraph Correction". In: *NIPS 2012 Workshop: Perturbations, Optimization, and Statistics* (2012).

[73] Z. Liu, Y. Zhang, E.Y. Chang, and M. Sun. "PLDA+: Parallel Latent Dirichlet Allocation with Data Placement and Pipeline Processing". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 2.3 (2011), p. 26.

[74] David JC MacKay. *Information Theory, Inference and Learning Algorithms.* Cambridge University Press, 2003.

[75] D.M. Malioutov, J.K. Johnson, and A.S. Willsky. "Walk-sums and Belief Propagation in Gaussian Graphical Models". In: *The Journal of Machine Learning Research* 7 (2006), pp. 2031–2064.

[76] Sean P Meyn and Richard L Tweedie. *Markov Chains and Stochastic Stability*. Cambridge University Press, 2009.

[77] M. J. Castel V. Migallón and J. Penadés. "On Parallel Two-Stage Methods for Hermitian Positive Definite Matrices with Applications to Preconditioning". In: *Electronic Transactions on Numerical Analysis* 12 (2001), pp. 88–112.

[78] K. Murphy. "Hidden semi-Markov models (segment models)". In: *Technical Report* (Nov. 2002). URL: http://www.cs.ubc.ca/~murphyk/Papers/segment.pdf.

[79] K. P. Murphy. *Conjugate bayesian analysis of the gaussian distribution*. Tech. rep. 2007.

[80] Kevin P Murphy. *Machine Learning: a Probabilistic Perspective*. MIT Press, 2012.

[81] Willie Neiswanger, Chong Wang, and Eric Xing. "Asymptotically Exact, Embarrassingly Parallel MCMC". In: *arXiv preprint arXiv:1311.4780* (2013).

[82] D. Newman, A. Asuncion, P. Smyth, and M. Welling. "Distributed Algorithms for Topic Models". In: *The Journal of Machine Learning Research* 10 (2009), pp. 1801–1828.

[83] D. Newman, A. Asuncion, P. Smyth, and M. Welling. "Distributed Inference for Latent Dirichlet Allocation". In: *Advances in Neural Information Processing Systems* 20.1081-1088 (2007), pp. 17–24.

[84] F. Niu, B. Recht, C. Ré, and S.J. Wright. "Hogwild!: A Lock-Free Approach to Parallelizing Stochastic Gradient Descent". In: *Advances in Neural Information Processing Systems* (2011).

[85] James R Norris. *Markov Chains*. 2008. Cambridge university press, 1998.

[86] Alan V. Oppenheim, Alan S. Willsky, and S. Hamid Nawab. *Signals and Systems*. Vol. 1. Prentice Hall, 1996, p. 957.

[87] P. Orbanz. "Construction of Nonparametric Bayesian Models from Parametric Bayes Equations". In: *Advances in Neural Information Processing Systems* (2009).

[88] Peter Orbanz. "Construction of Nonparametric Bayesian Models from Parametric Bayes Equations." In: *NIPS*. 2009, pp. 1392–1400.

[89] G. Papandreou and A. Yuille. "Gaussian Sampling by Local Perturbations". In: *Neural Information Processing Systems (NIPS)*. 2010.

[90] A. Parker and C. Fox. "Sampling Gaussian Distributions in Krylov Spaces with Conjugate Gradients". In: *SIAM Journal on Scientific Computing* 34.3 (2012), pp. 312–334.

[91] Andrew N Pressley. *Elementary Differential Geometry*. Springer, 2010.

[92] Rajesh Ranganath, Chong Wang, Blei David, and Eric Xing. "An Adaptive Learning Rate for Stochastic Variational Inference". In: *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*. 2013, pp. 298–306.

[93] Herbert Robbins and Sutton Monro. "A Stochastic Approximation Method". In: *The Annals of Mathematical Statistics* (1951), pp. 400–407.

[94] Christian P Robert and George Casella. *Monte Carlo Statistical Methods*. Vol. 319. Citeseer, 2004.

[95] Gian-Carlo Rota and W Strang. "A Note on the Joint Spectral Radius". In: (1960).

[96] Nicholas Ruozzi and Sekhar Tatikonda. "Message-Passing Algorithms for Quadratic Minimization". In: *Journal of Machine Learning Research* 14 (2013), pp. 2287–2314. URL: http://jmlr.org/papers/v14/ruozzi13a.html.

[97] Martin J Russell and A Cook. "Experimental Evaluation of Duration Modelling Techniques for Automatic Speech Recognition". In: *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'87*. Vol. 12. IEEE. 1987, pp. 2376–2379.

[98] Martin Russell and Roger Moore. "Explicit Modelling of State Occupancy in Hidden Markov Models for Automatic Speech Recognition". In: *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'85*. Vol. 10. IEEE. 1985, pp. 5–8.

[99] Ardavan Saeedi and Alexandre Bouchard-Côté. "Priors over Recurrent Continuous Time Processes". In: *NIPS*. Vol. 24. 2011, pp. 2052–2060.

[100] Odette Scharenborg, Vincent Wan, and Mirjam Ernestus. "Unsupervised Speech Segmentation: an Analysis of the Hypothesized Phone Boundaries". In: *The Journal of the Acoustical Society of America* 127.2 (2010), pp. 1084–1095.

[101] D. Serre. Nov. 2011. URL: http://mathoverflow.net/questions/80793/is-gauss-seidel-guaranteed-to-converge-on-semi-positive-definite-matrices/80845#80845.

[102] J. Sethuraman. "A constructive definition of Dirichlet priors." In: *Statistica Sinica*. Vol. 4. 1994, pp. 639–650.

[103] Jayaram Sethuraman. "A Constructive Definition of Dirichlet Priors". In: *Statistica Sinica* 4 (1994), pp. 639–650.

[104] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. "Practical Bayesian Optimization of Machine Learning Algorithms". In: *arXiv preprint arXiv:1206.2944* (2012).

[105] Eduardo D Sontag. *Mathematical Control Theory: Deterministic Finite Dimensional Systems*. Vol. 6. Springer, 1998.

[106] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. "Hierarchical Dirichlet Processes". In: *Journal of the American Statistical Association* 101.476 (2006), pp. 1566–1581.

[107] Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. "Sharing Clusters Among Related Groups: Hierarchical Dirichlet Processes". In: *NIPS*. 2004.

[108] S. E. Tranter and D. A. Reynolds. "An Overview of Automatic Speaker Diarization Systems". In: *Audio, Speech, and Language Processing, IEEE Transactions on* 14.5 (2006), pp. 1557–1565.

[109] D. A. Van Dyk and X. L. Meng. "The Art of Data Augmentation". In: *Journal of Computational and Graphical Statistics* 10.1 (2001), pp. 1–50.

[110] J. Van Gael, Y. Saatci, Y. W. Teh, and Z. Ghahramani. "Beam Sampling for the Infinite Hidden Markov Model". In: *Proceedings of the 25th International Conference on Machine Learning*. ACM. 2008, pp. 1088–1095.

[111] Sabine Van Huffel and Vasile Sima. "SLICOT and Control Systems Numerical Software Packages". In: *Proc. 2002 IEEE Int. Conf. Control Applications and IEEE Int. Symp. Computer Aided Control System Design, CCA/CACSD 2002*. 2002, pp. 39–44.

[112] Sabine Van Huffel, Vasile Sima, Andras Varga, Sven Hammarling, and François Delebecque. "High-performance Numerical Software for Control". In: *IEEE Control Systems Magazine* 24.1 (2004), pp. 60–76.

[113] M. J. Wainwright and M. I. Jordan. "Graphical Models, Exponential Families, and Variational Inference". In: *Foundations and Trends® in Machine Learning* 1.1-2 (2008), pp. 1–305.

[114] Chong Wang and David Blei. "Truncation-free Online Variational Inference for Bayesian Nonparametric Models". In: *Advances in Neural Information Processing Systems 25*. 2012, pp. 422–430.

[115] Chong Wang, John W Paisley, and David M Blei. "Online Variational Inference for the Hierarchical Dirichlet Process". In: *International Conference on Artificial Intelligence and Statistics*. 2011, pp. 752–760.

[116] Max Welling and Yee W Teh. "Bayesian Learning via Stochastic Gradient Langevin Dynamics". In: *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*. 2011, pp. 681–688. URL: http://machinelearning.wustl.edu/mlpapers/paper_files/ICML2011Welling_398.pdf.

[117] Tianbing Xu and Alexander T Ihler. "Multicore Gibbs sampling in Dense, Unstructured Graphs". In: *International Conference on Artificial Intelligence and Statistics*. 2011, pp. 798–806.

[118] S. Z. Yu. "Hidden Semi-Markov Models". In: *Artificial Intelligence* 174.2 (2010), pp. 215–243.

[119] M. Zeifman and K. Roth. "Nonintrusive Appliance Load Monitoring: Review and Outlook". In: *Consumer Electronics, IEEE Transactions on* 57.1 (2011), pp. 76–84.